Dissertation

Mobility Improves the Security of Location Awareness in Wireless Networks

Thesis approved by the Department of Computer Science
of the University of Kaiserslautern (TU Kaiserslautern)
for the award of the Doctoral Degree

Doctor of Engineering (Dr.-Ing.)

to

Matthias Schäfer

Date of Defense:	June 29, 2018
Dean:	Prof. Dr. Klaus Schneider
PhD Committee	
Chair:	Prof. Dr. Heike Leitte
Reviewers:	Prof. Dr. Jens B. Schmitt Prof. Dr. Christina Pöpper Dr. Vincent Lenders

D 386

Abstract

Mobility has become an integral feature of many wireless networks. Along with this mobility comes the need for location awareness. A prime example for this development are today's and future transportation systems. They increasingly rely on wireless communications to exchange location and velocity information for a multitude of functions and applications. At the same time, the technological progress facilitates the widespread availability of sophisticated radio technology such as software-defined radios. The result is a variety of new attack vectors threatening the integrity of location information in mobile networks.

Although such attacks can have severe consequences in safety-critical environments such as transportation, the combination of mobility and integrity of spatial information has not received much attention in security research in the past. In this thesis we aim to fill this gap by providing adequate methods to protect the integrity of location and velocity information in the presence of mobility. Based on physical effects of mobility on wireless communications, we develop new methods to securely verify locations, sequences of locations, and velocity information provided by untrusted nodes. The results of our analyses show that mobility can in fact be exploited to provide robust security at low cost.

To further investigate the applicability of our schemes to real-world transportation systems, we have built the OpenSky Network, a sensor network which collects air traffic control communication data for scientific applications. The network uses crowdsourcing and has already achieved coverage in most parts of the world with more than 1000 sensors.

Based on the data provided by the network and measurements with commercial off-the-shelf hardware, we demonstrate the technical feasibility and security of our schemes in the air traffic scenario. Moreover, the experience and data provided by the OpenSky Network allows us to investigate the challenges for our schemes in the real-world air traffic communication environment. We show that our verification methods match all requirements to help secure the next generation air traffic system.

Acknowledgements

I thank Prof. Jens Schmitt, for always giving great and thoughtful advice, for being indescribably supportive, and for his altruistic, warm, and friendly way to accompany his students through the ups and downs of becoming a PhD.

I thank Dr. Vincent Lenders, for sharing with me his enthusiasm, creativity, commitment, and friendship which ultimately made this thesis (and a lot more great things) possible.

I thank Markus Fuchs, for being an awesome and reliable friend and companion in literally all of my endeavors and for always providing a sympathetic ear.

I thank my longtime collaborators Dr. Martin Strohmeier and Prof. Ivan Martinovic for learning and growing together, for sharing your ideas and extraordinary abilities with me, and for all the fun and interesting times we have had over the years.

I thank my awesome SeRo Systems crew Michael Arndt, Markus Engel, Markus Fuchs, and Marco Meides for enduring the endless yet fruitful discussions and for helping me realize my dreams.

I thank my colleagues at DISCO (Dr. Daniel Berger, Dr. Michael Beck, Dr. Steffen Bondorf, Barbara Erlewein, Anja Gerber, Paul Nikolaus, Carolina Nogueira, Dr. Wint Yi Poe, Dr. Hao Wang, Dr. Matthias Wilhelm) for all the laughter and fun we had and for the cheerful environment you provided throughout my PhD.

I thank Prof. Christina Pöpper for showing such interest in my research and for taking the time to think about it and provide valuable feedback.

Last but certainly not least, I thank my parents (Heike and Dieter), my sisters and their families (Juliane, Jan, Josephine and Joshua and Judith and Olaf), my “extended family” (Verena, Markus E., Ferrah, Tim and Leo), and all my valuable friends for their love, friendship, joy, support, patience, honesty, and safety which gave me the strength I needed to reach this goal.

Contents

List of Figures xii

List of Tables xiii

List of Abbreviations xv

1 Introduction 1

1.1 Motivation 2
1.2 Research Contributions 3
1.3 Thesis Outline 5

2 Location Awareness, Mobility & Security 7

2.1 Mobility in Networks 8
  2.1.1 Location Awareness 8
  2.1.2 Examples 9
  2.1.3 System Model & Notation 11
2.2 Location Verification 14
  2.2.1 Existing Methods 17
  2.2.2 Challenges in Mobile Networks 20
2.3 Effects of Mobility on Wireless Transmissions 22
  2.3.1 Time Domain 22
  2.3.2 Frequency Domain 24
  2.3.3 Space Domain 27

3 Improving Location Verification with Mobile Verifiers 29

3.1 Network & Threat Model 30
3.2 Verification Procedure 31
  3.2.1 Directed Mobility vs. Mobility of Opportunity 32
  3.2.2 Protocol Properties 33
3.3 Security Analysis 34
  3.3.1 Directed Mobility 38
  3.3.2 Mobility of Opportunity 40
  3.3.3 Movement Patterns 46
  3.3.4 Lessons Learned 48

3.4 Discussion . . . 49
3.4.1 Stronger Adversaries . . . 50

4 Verification of Mobile Provers 53
4.1 Network & Threat Model . . . 54
4.2 Verifying Tracks in the Time Domain . . . 58
4.2.1 Security Analysis . . . 59
4.2.2 Dealing With Noise . . . 65
4.2.3 Evaluation . . . 69
4.2.4 Realistic Movement Patterns . . . 75
4.2.5 Discussion . . . 81
4.3 Verifying Motion in the Frequency Domain . . . 84
4.3.1 Security Analysis . . . 85
4.3.2 Evaluation . . . 88
4.3.3 Adapted Verification Scheme . . . 92
4.3.4 Real-world Performance . . . 98
4.3.5 Discussion . . . 100
4.4 Outlook: Verifying Mobility in the Space Domain . . . 101
4.4.1 Sanity Check Protocol . . . 102
4.4.2 Lossy Communication Channel . . . 104
4.4.3 Security Considerations . . . 105

5 Air Traffic Surveillance Scenario 107
5.1 Automatic Dependent Surveillance-Broadcast . . . 108
5.1.1 Technical Background . . . 110
5.1.2 (In-)Security of ADS-B . . . 111
5.2 The OpenSky Network . . . 112
5.2.1 A Brief History . . . 112
5.3 Challenges for Track & Motion Verification . . . 114
5.3.1 1090 MHz Frequency Overuse . . . 114
5.3.2 ADS-B Data Quality . . . 118
5.3.3 Scalability . . . 121
5.3.4 Crowdsourcing Security . . . 122

6 Summary & Future Work 129
6.1 Key Results . . . 130

6.2 Future Directions . . . 132
6.3 Final Conclusions . . . 132

Bibliography **135**

Curriculum Vitae **143**

List of Figures

2.1 Mobility-differentiated Time of Arrival Principle & Notation 24
2.2 Doppler-shift Principle & Notation 25

3.1 Example attack on a single verifier in the time domain 35
3.2 Example with two verifiers and the resulting time domain constraints . . 38
3.3 Example for secure verification with directed movement 39
3.4 Notation and movement model used for our simulations 42
3.5 Results of our simulations for different step widths/speeds 44
3.6 Number of intersections of 10,000 random movements 45
3.7 Example scenario resulting in 6 intersections of the constraints. 46
3.8 Results of our simulations investigating relative movements of two verifiers 47

4.1 Example track verification scenarios with three verifiers 63
4.2 Clock drifts of 8 OpenSky Network receivers over 1 hour 71
4.3 ECDF of simulated track verification results 73
4.4 Local track verification performance for different parameters 74
4.5 Results of the global track verification scheme for 1000 simulated tracks . 75
4.6 The tracks from the OpenSky Network used for our simulations 76
4.7 Time to verify real aircraft trajectories 78
4.8 False acceptance/rejection rate for real aircraft trajectories 79
4.9 Illustration of the linear track problem 79
4.10 Effect of linear trajectories on the verification result 80
4.11 The effect of linear tracks on security 81
4.12 Example for a scenario with two verifiers not verifyable by our scheme . . 86
4.13 The tracks recorded by two USRPs for our evaluation 89
4.14 Results of our frequency of arrival measurements 91
4.15 Results of our frequency-difference of arrival measurements 96
4.16 Q-Q plot comparing the measured values to our error model 96
4.17 Locations and ranges of the three OpenSky receivers used for our analysis 98
4.18 OpenSky data used for our the motion verification feasibility study . . . 99
4.19 Results of our simulations for different thresholds 99
4.20 The results using two successive motion claims 100
4.21 Example attack scenario in the space domain 103

5.1 Modernization of civil air traffic surveillance 109
5.2 1090ES ADS-B Modulation and Coding 110
5.3 Growth of OpenSky's data set over time 112
5.4 Coverage of the OpenSky Network 113
5.5 Results of our ADS-B message loss analysis. 117
5.6 Results of our ADS-B accuracy analysis. 120

List of Tables

2.1 Symbols and Notations Overview . . . 15
2.2 Location Verification Methods Comparison . . . 21

4.1 Track verification simulation parameters . . . 70
4.2 Results of the frequency of arrival-based measurements . . . 91
4.3 Results of the final frequency-difference of arrival measurements . . . 95

5.1 Duration and rate of SSR Mode A/S, S, and ADS-B transmissions. . . . 116
5.2 Surveillance Integrity Level reported with ADS-B . . . 119

List of Abbreviations

ADS-B	Automatic Dependent Surveillance-Broadcast
AIS	Automatic Identification System
AOA	Angle of Arrival
ATC	Air Traffic Control
CPS	Cyber-physical Systems
DOP	Dilution of Precision
EPU	Estimated Position Uncertainty
ES	Extended Squitter
ETCS	European Train Control System
FAA	Federal Aviation Administration
FDOA	Frequency Difference of Arrival
FFT	Fast Fourier Transform
FOA	Frequency of Arrival
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPSDO	GPS-Disciplined, Oven-controlled Crystal Oscillator
HCR	Horizontal Containment Radius
ICAO	International Civil Aviation Organization
MDTOA	Mobility-differentiated Time of Arrival
NAC	Navigation Accuracy Category
NextGen	Next Generation Air Transportation System

NIC	Navigation Integrity Category
PGP	Pretty Good Privacy
RMSE	root-mean-square error
SESAR	Single European Sky ATM Research
SIL	Source Integrity Level
SSR	Secondary Surveillance Radar
TDOA	Time Difference of Arrival
TISB	Traffic Information Service-Broadcast
TOA	Time of Arrival
TOF	Time of Flight
UAT	Universal Access Transceiver

What makes a problem a problem is not that a large amount of search is required for its solution, but that a large amount would be required if a requisite level of intelligence were not applied.

— Allen Newell and Herbert A. Simon [1]

1

Introduction

Contents

1.1 Motivation 2
1.2 Research Contributions 3
1.3 Thesis Outline 5

1.1 Motivation

Since the Global Positioning System (GPS) has been opened to the public to its full extent in May 2000, location awareness has become the foundation of many systems and applications. The nearly global coverage and the availability of cheap and accurate GPS receivers have enabled many location-based applications including, but certainly not limited to, augmented reality games, navigation systems, emergency localization, and tracking of individuals, vehicles, or animals. One of the domains most affected by this development is transportation. While GPS-based navigation has already become an integral part of our transportation system, future intelligent transportation systems are expected to use GPS data provided by vehicles to analyze, predict, and control traffic behavior [2]. Moreover, autonomous cars are expected to be aware of each other's exact locations as well as their speed and direction of movement. Having this information, cars can form tightly spaced platoons to increase the highway capacity and reduce fuel consumption and carbon dioxide emission [3].

While some parts of such intelligent transportation systems might still be some ways off in the future, an increasing level of autonomy of vehicles is in fact a trend observed in all modes of transportation. It is driven by the prospect of an increase in efficiency and capacity, a reduction of labor costs, and better safety by removing the risk of human error. Car makers such as Mercedes-Benz and Tesla as well as other companies such as Google and Uber are putting enormous efforts into the development of self-driving cars and trucks. In aviation, unmanned aerial vehicles and manned instrument-based flights are already relying on automatic flight systems for most of the time [4]. Even driver-less ships and trains are currently finding their way into transportation of passengers and cargo [5,6].

A basic premise for safe autonomy, however, is a complete situational awareness which includes the locations and movements of nearby vehicles. This knowledge is required by critical functions such as collision avoidance and traffic surveillance. A trending approach to accomplish location awareness is to have vehicles sharing their exact positions and velocities using wireless technology, rather than having dedicated infrastructure locating them. For example, the next generation air transportation surveillance technology Automatic Dependent Surveillance-Broadcast (ADS-B), which is currently being deployed worldwide, enables aircraft to periodically broadcast their GPS positions and velocities to nearby aircraft and surveillance infrastructure on the ground. Another example is the

Automatic Identification System (AIS) used by ships to share their GPS position with other ships and satellite-based surveillance systems.

This simple approach has many advantages, ranging from lower costs by using existing positioning infrastructure to better coverage and flexibility as most existing localization techniques require a direct line of sight to the target. From a security perspective, however, this approach is highly problematic. Without further means to verify the accuracy of reported locations and velocities, users of this information are forced to trust the source blindly. In fact, the literature describes many attacks based on abusing this trust and violating the integrity of reported location information, including attacks on ADS-B [7] and vehicular ad hoc networks [8–10]. Moreover, several studies have demonstrated that attacks based on injecting false ADS-B information are favored by the widespread availability of software-defined radios, making them cheap and easy to launch at the same time [11–13].

The sobering consequence of these findings is that the safety of today's and future transportation systems is compromised as long as these issues are not addressed appropriately. Whether automatic or manual, decisions based on false information can have severe consequences, especially in safety-critical domains such as transportation. It is therefore crucial to protect their integrity by implementing methods to securely verify location and velocity information. However, the nature of the transportation environment renders many existing solutions useless. As mentioned by Parno and Perrig in [8], mobility is the norm in such systems whereas traditional security research frequently assumes relatively static networks. In addition to that, the constant cost pressure and the organizational complexity of the transportation domain prevent the use of expensive or active solutions. The result of all this is a lack of adequate methods that take both the challenges arising from mobility as well as the peculiarities of the transportation ecosystem into account.

This thesis aims to fill this gap by proposing new verification approaches that are specifically designed to meet the demanding requirements of the transportation domain. Our goal is to find ways to deal with mobility and to provide realistic methods to protect the integrity of location and velocity information in mobile networks.

1.2 Research Contributions

This section briefly summarizes the contributions to the research community on which this thesis is based on.

- Our interest in verification of location information in mobile networks was initially sparked by our **Experimental Analysis of Attacks on Next Generation Air Traffic Communication** [13], published on the *11th International Conference on Applied Cryptography and Network Security.* This study investigates the feasibility and requirements of attacks on ADS-B. Its results were the primary motivation for the verification of mobile provers considered in chapter 4.

- The idea to address and exploit mobility by incorporating it into the design has first been presented at the *36th IEEE Symposium on Security and Privacy* under the title **Secure Track Verification** [18]. This paper includes large parts of the fundamental background provided in chapter 2 and the time domain verification of tracks provided in section 4.2. The space domain verification protocol provided in section 4.4 is also part of this paper. The paper was awarded by armasuisse with 1st place *"Cyber Award" for outstanding scientific contributions* in 2016.

- The rest of the foundations in chapter 2 and the extension of the previous idea to the frequency domain of signals (section 4.3) has been published on the *9th ACM Conference on Security & Privacy in Wireless and Mobile Networks* in our paper **Secure Motion Verification** [19]. The paper was awarded by armasuisse with 3rd place *"Cyber Award" for outstanding scientific contributions* in 2017.

- Following the analysis of attacks on ADS-B, a broader investigation of challenges in the air traffic scenario has been published in our article **Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B** [14] in the *IEEE Communications Magazine.* This article served as a basis for the analysis of the 1090 MHz frequency overuse in subsection 5.3.1.

- A large-scale analysis of the ADS-B environment has been published at the *35th IEEE/AIAA Digital Avionics Systems Conference* under the title **OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage** [17]. Parts of these results are presented in the ADS-B data quality analysis in subsection 5.3.2.

- To gather data for our research, we have built the OpenSky Network in collaboration with researchers from armasuisse (Switzerland) and the University of Oxford (United Kingdom). The network was first presented to the public at the *13th IEEE/ACM International Symposium on Information Processing in Sensor Networks* in our paper **Bringing Up OpenSky: A Large-scale ADS-B Sensor**

Network for Research [15] and in a demo with the same title [16]. The demo won the best demo award 1st runner up. Since then, the network has evolved into an open research platform providing data to different research fields. As of this writing, the network has contributed with its data to over 20 publications and continues its operations as a non-profit association. The insights on crowdsourcing provided in subsection 5.3.4 are based on our experiences with the network.

- The trust model for OpenSky's data described in subsection 5.3.4 is part of our paper **Crowdsourcing Security for Wireless Air Traffic Communications** [20] which has been published on the *9th International Conference on Cyber Conflict.*

1.3 Thesis Outline

The remainder of this thesis is structured as follows:

- **Chapter 2** provides the overall system model and notation used throughout this thesis. It also provides an overview and comparison of the state of the art of location verification and summarizes the effect of mobility on wireless communications.

- **Chapter 3** proposes a location verification scheme based on mobile verifiers. The scheme exploits the effect of mobility on wireless signals in the time domain to provide robust security with low system requirements. Our formal analysis proves the security of the scheme and we investigate the impact of different movement patterns on the security of the scheme in extensive simulations.

- **Chapter 4** shows how to incorporate the mobility of devices into the verification problem. We define track and motion verification and provide a solution to each of these problems. The feasibility of both schemes is demonstrated in a mix of simulations and experiments in the air traffic scenario.

- **Chapter 5** investigates the challenges our new schemes have to face in a real-world ADS-B environment. We discuss solutions to each of the problems. In addition, the section includes an in-depth discussion of the adequacy of crowdsourcing as a basis for security services.

- **Chapter 6** provides an overview of the key results of this thesis and discusses future work.

To say of what is, that it is, or of what is not, that it is not, is true.

— Aristotle in Metaphysics (Book 4)

2

Location Awareness, Mobility & Security

Contents

2.1 Mobility in Networks **8**
  2.1.1 Location Awareness 8
  2.1.2 Examples 9
  2.1.3 System Model & Notation 11
2.2 Location Verification **14**
  2.2.1 Existing Methods 17
  2.2.2 Challenges in Mobile Networks 20
2.3 Effects of Mobility on Wireless Transmissions **22**
  2.3.1 Time Domain 22
  2.3.2 Frequency Domain 24
  2.3.3 Space Domain 27

2.1 Mobility in Networks

While the combination of the terms “mobile” and “network” is often associated with cellular networks, the system model considered in this work is rather motivated by modern Cyber-physical Systems (CPS). In particular, we concentrate on CPS in which the locations of physical system components are essential to the system’s applications. Many of such systems form the backbone of today’s critical infrastructures, including transportation, environmental monitoring, emergency services, or defense to name but a few. Inherent to these systems is that the mobility of components makes continuous location awareness a crucial and challenging requirement. For instance, the coordination of security and emergency personnel during large events requires the central management to constantly track the whereabouts of its units to be able to effectively control their actions.

As the nature of this example suggests, this work is not focused on *node-local* location awareness in which a user, or more generally, a network node, wants to know where she or he is. Although node-local location awareness is also relevant as it is used by many of today’s everyday applications such as navigation or tracking of fitness activities, this work concentrates on a more external model of location awareness in which the nodes or some global entity in the network rely on knowledge about other nodes’ locations. More specifically, we focus on the security challenges that arise from both this dependence on external information and the mobility of nodes. The following sections will refine this system model, provide real-world examples, and introduce the formal foundations and background needed to analyze the interplay of mobility, location awareness, and security.

2.1.1 Location Awareness

In order to establish continuous location awareness while facing constant and rapid movements, wireless communication technologies are essential to determine and exchange current positions of nodes in the system. One approach to *tracking* locations of nodes is to use centralized localization techniques such as trilateration or radars. An overview of secure localization techniques is provided in [21]. A major characteristic of this approach is that the nodes themselves are not aware of their own location per se. If required, the localization infrastructure has to feed the locations back to the nodes. In practice, however, this approach often comes with major drawbacks such as limited scalability, costs, and physical constraints, for example, due to a required line of sight between nodes and localization infrastructure. As a consequence of these drawbacks and fueled by the proliferation of satellite-based positioning systems, an increasingly popular approach is to enable mobile nodes in the network to determine their exact locations themselves by equipping them with positioning devices such as GPS receivers. The locations are then shared with the network over wireless communication links. The latter point is crucial as it provides the global knowledge that enables applications to act beyond node-local location awareness.

We conclude the following architectural system properties that arise from this model of location awareness and are used in this thesis:

- *Positioning infrastructure:* We assume that nodes are able to use a positioning infrastructure which allows them to determine their own location and motion parameters.
- *Remote tracking:* We consider scenarios and applications where nodes and network functions rely on other nodes' locations and motion.
- *Discrete updates:* Remote tracking requires that nodes periodically or sporadically share their locations. As the communication channel is a constraint resource, update rates are usually limited and we obtain a discrete location model.

2.1.2 Examples

One domain in which this paradigm of location awareness has become the foundation for many essential services are today's and future intelligent transportation systems. Core functions, such as traffic surveillance, traffic management, autopilots, and collision avoidance are all based on a continuous and accurate location awareness and the integrity of this location awareness is essential for safe operation. Unfortunately, most of these systems have evolved over long periods of time and lack the flexibility to adapt to the changing threat landscape fostered by the rapid technological progress. As a result, they are vulnerable to a variety of attacks, mostly based on malicious violations of the integrity of the location awareness. In the following, we will outline a few examples of systems used in transportation which match this profile.

- Rail transport: In the European Train Control System (ETCS) (Level 2 and 3), trains determine their locations using trackside beacons (balises) and periodically report their exact location to a central control system [22]. This control system can then use this information for line speed enforcement, speed regulation, train separation, and collision avoidance. Security measures to ensure the integrity of information are not part of the specification, which renders this and similar systems vulnerable to attacks that can compromise safety and cause serious accidents [23].

- Maritime transport: The AIS is a system used to signal a vessel's identification, location, course, and speed to other vessels and authorities [24]. Location and speed are usually determined using a Global Navigation Satellite System (GNSS) such as GPS and periodically broadcasted to receivers in range. As in the case of ETCS, security is not part of the specification of AIS and the system is vulnerable to a number of attacks, many of which based on the violation of the integrity of the location awareness of vessels and authorities [25].

- Air transport: The Next Generation Air Transportation System (NextGen) (more details in section 5.1) is built on a system named ADS-B. Similar to AIS, the transmitting subsystem of ADS-B enables aircraft to periodically broadcast their GNSS-based locations and velocities [26]. This information is received by ground stations and other aircraft and used for many applications including collision avoidance, traffic management, or commercial tracking websites. This system also joins the ranks of ETCS and AIS as it does not implement any security measures. As the other systems, ADS-B is vulnerable to several attacks based on violations of the integrity of location awareness [11, 13].

- Road transport: In future intelligent transportation systems, vehicles on the roads are envisioned to exchange location and movement information to form tightly spaced platoons in order to increase the highway capacity [3]. Recent research, however, has shown that attackers might be able to provoke life-threatening situations by tampering with the exchanged location information [10].

Due to their criticality and vulnerability, the security considerations, system models, constraints, and protocols designed and investigated in this thesis are mostly motivated by these transportation systems. However, it is worth noting that their applicability goes beyond transportation systems. The results of this work equally hold for any other system in which mobility and location awareness come together and form the basis of core functions. For example, mobile ad hoc networks may rely on location awareness to maximize routing efficiency or ensure a certain quality of service [27]. However, without protection of the integrity of location information shared across the network, applications are vulnerable to attacks such as sinkhole attacks or Sybil attacks [28].

2.1.3 System Model & Notation

Throughout this thesis, we will use the following notations and terminology. The scope of this work is limited to those parts of systems where location information is exchanged, that is, the wireless communication part. We generally refer to these parts as *networks*. The senders and receivers in such a network are collectively called *nodes*. Networks or parts of networks are denoted by calligraphic uppercase symbols and single nodes are denoted by ordinary capital letters. Following this convention, let $\mathcal{N} = \{N_1, \ldots, N_n\}$ be a network consisting of n nodes.

A Node's Spatial State

Since node locations and movements play a central role in this thesis, each node N_i is associated with a spatial state consisting of its *location* and *velocity*. In order to capture the mobility exhibited by some or all nodes in the network, both locations and velocities may be subject to changes and are therefore expressed as functions of time. Let $p_{N_i}(t)$ denote the location of some node $N_i \in \mathcal{N}$ at time t expressed as a vector in a Cartesian coordinate system. Whenever a node's identity is clear from its index in a specific context, we omit the uppercase node symbol and simply write $p_i(t)$. Analogously, velocities are denoted by $v_{N_i}(t)$ or, respectively, $v_i(t)$. Similar to locations, velocities are Cartesian vectors with the same number of dimensions. They combine a node's *speed* (position change rate) and its *direction* of movement. More specifically, if at time t_1 the node N_i is located at $p_i(t_1)$ and is moving with velocity $v_i(t_1)$ and does not change its speed and direction until time t_2, i.e. $v_i(t) = v_i(t_1)$ for $t_1 \leq t \leq t_2$, its location at each of these points in time t is

$$p_i(t) = p_i(t_1) + v_i(t_1) \cdot (t - t_1) \ .$$

We call the Euclidean norm $\|v_i(t)\|$ the speed of N_i at time t and the direction of N_i's movement at time t is expressed by the unit vector $v_i(t)/\|v_i(t)\|$. If N_i is *stationary*, its velocity is the zero vector at all times, i.e., $v_i(t) = \vec{0}$ for all t. In contrast, a node is *mobile* if there exists a point in time t during the considered period of time where $v_i(t) \neq \vec{0}$. It is worth noting that our model of mobility does not require mobile nodes to move constantly, i.e., they might be temporarily stationary.

Global Time vs. Local Timestamp

We distinguish between global points in time and local timestamps. Time values (denoted by t) mark globally unique and absolute points in time referring to a perfect global clock. On the contrary, timestamps (denoted by $\hat{t}$) are values of simple node-local counters clocked by local oscillators. Local clocks do not have a well-defined reference time. We assume, however, that oscillator speeds are known and timestamps can be converted to a common unit (e.g. nanoseconds).

This concept of local timestamps and global times is comparable to having a simple stopwatch (local counter) and a GPS-synchronized atomic clock (global clock). The stopwatch might have been started at any arbitrary point in time and without further means of synchronization, the values shown on the display (timestamps) can only be used to measure *periods* of time, but not exact *points* in time. This is a crucial difference because having global clocks allows measuring periods of times with time values from different devices since they are all measured using the same time reference. In contrast, local timestamps from different devices cannot be related to each other without knowing their real offset. For example, Alice and Bob want to measure the duration of a flight from Frankfurt International to London Heathrow while Alice is located in Frankfurt and Bob is located in London. If they both have access to a global clock, they can simply exchange the clock values observed at the departure and arrival of the flight and calculate their difference to get the flight duration. If they only have stopwatches (local clocks), the difference between the timestamps observed by Alice and Bob does not provide any information about the duration of the flight. To measure the actual duration using their stopwatches, they will either have to synchronize the point in time when they start their stopwatches or one of them has to be on the flight and measure both departure and arrival with the same stopwatch.

We call nodes with access to a network-global clock *synchronized* and nodes with simple local counters *unsynchronized*. As illustrated by the previous example, being synchronized offers on the one hand great advantages for applications acting across multiple nodes. On the other hand, however, synchronization is costly as it requires either additional communication to synchronize the nodes or additional infrastructure and hardware to access a network-global clock. Both is particularly problematic in scenarios with cost and resource limitations. *We therefore assume that nodes in our system are not synchronized.* This assumption improves the flexibility and applicability of our results while keeping the system requirements and the costs low.

Let t_1 and t_2 be two points in time. Let further $\hat{t}_{1,A}$, $\hat{t}_{2,A}$, $\hat{t}_{1,B}$, and $\hat{t}_{2,B}$ be the respective timestamps measured by two unsynchronized nodes A and B at t_1 and t_2 respectively. Then the relationship

$$\hat{t}_{2,A} - \hat{t}_{1,A} = \hat{t}_{2,B} - \hat{t}_{1,B} = t_2 - t_1$$

holds according to our model and there is an unknown offset Δ_t between the local clocks:

$$\Delta_t = \hat{t}_{1,A} - \hat{t}_{1,B} = \hat{t}_{2,A} - \hat{t}_{2,B} \ .$$

We assume that the offset Δ_t is constant during the considered time periods. We will loosen this assumption later by adding an error model including measurement noise and clock drift to this basic notion of time.

Location Awareness

To formalize the concept of location awareness, we distinguish between nodes $\mathcal{R} \subseteq \mathcal{N}$ that wish to be aware of other nodes' spatial states, i.e., the receiving side of the information, and nodes $\mathcal{S} \subseteq \mathcal{N}$ whose spatial states are of interest, i.e., the sending side of information. Is is worth noting here that $\mathcal{R}$ and $\mathcal{S}$ can, but do not have to be disjoint sets of nodes. In a pure traffic surveillance scenario, for example, $\mathcal{R}$ might consist of base stations placed alongside highways which receive location information from cars $(= \mathcal{S})$ and forward them to a central surveillance system. In this case, $\mathcal{R}$ and $\mathcal{S}$ would be disjoint sets of nodes, i.e., $\mathcal{R} \cap \mathcal{S} = \emptyset$. In contrast, in a pure collision avoidance system, cars would exchange their spatial state in a peer-to-peer-like fashion and thus, $\mathcal{R} = \mathcal{S}$. Many real-world systems, however, are a mix of both scenarios, i.e., $\mathcal{R} \cap \mathcal{S} \neq \emptyset$, $\mathcal{R} \not\subset \mathcal{S}$, and $\mathcal{S} \not\subset \mathcal{R}$ may all be true for the same network. Both sets $\mathcal{S}$ and $\mathcal{R}$ might also change over time. If, for instance, an aircraft leaves the airspace managed by a service of interest, it will be removed from $\mathcal{S}$ as its spatial state becomes irrelevant to the service's operation.

In order to establish location awareness, all nodes in $\mathcal{S}$ broadcast their spatial state to all nodes in $\mathcal{R}$. In particular, a node N_i broadcasts messages $m_i = (i, \hat{t}, p_i(t), v_i(t))$, where time t refers to the transmission time of the message and $\hat{t}$ is a sender-local timestamp referring to t. Transmissions can be triggered either periodically, on request, or event-based (e.g., on location changes). The update rate and resolution depends on the requirements of the applications and the network's mobility pattern. Aircraft will likely broadcast their locations and velocities more often than ships, since aircraft typically move at much higher speeds. In fact, location and velocity are both broadcasted by aircraft using ADS-B twice per second, while AIS provides update rates between 2 seconds for fast and 3 minutes for anchored vessels.

Communication Medium

We assume that nodes use a typical wireless communication medium such as electromagnetic waves (radio frequency) or ultrasound to exchange messages. For the purpose of this thesis, the communication medium is specified only by its center frequency f_0 and the signal propagation speed c. We assume that all senders regularly transmit their signals at f_0. We assume that the only non-negligible delay experienced during transmissions is the propagation delay δ. More specifically, a message m_i sent by N_i at time t is received by node $R \in \mathcal{R}$ at time

$$t_R = t + \delta = t + \|p_i(t) - p_R(t_R)\|/c \ ,$$

where $\| \cdot \|$ denotes the Euclidean norm. In networks where radio communication is used, i.e., $c \approx 299792458$ meters per second, we assume that $\|v_i(t)\| \ll c$ and simplify the above relationship by approximating $p_R(t_R) \approx p_R(t)$. In fact, this assumption is valid for almost all real-world systems since the much slower ultrasound medium is usually not used in systems which exhibit high speed movements due to its limited range.

An overview of the notations and symbols introduced in this and the following paragraphs is provided in Table 2.1 on page 15.

2.2 Location Verification

It is obvious that networks which make a certain effort to establish location awareness have the wish to do so reliably. Especially in critical applications such as transportation or location-based access control, ensuring the integrity of a node's claimed location is inevitable to guarantee safe and secure operation. Classic cryptography alone can only provide information security in terms of confidentiality and integrity during the exchange of information. However, it does not provide means to verify the truthfulness of the information itself. Motivated by this need, researchers have proposed many approaches over the last 25 years to securely verify location information from untrusted nodes in a network.

The general problem addressed by this rich body of work is called *location verification*. It has been defined by Sastry et al. in [29] as follows: "A set of *verifiers* $\mathcal{V}$ wish to verify

Symbol	Description
$\mathcal{A}, \mathcal{B}, \ldots, \mathcal{Z}$	Sets of nodes forming a network or part of a network.
$A, B, \ldots, Z, N_i, i$	Single network nodes. They might have a unique index (subscript) to indicate that nodes belong to the same group of nodes (e.g. $N_1, N_2, \ldots$). Whenever the context allows a unique mapping between index and node, we will simply use the index to refer to the node.
t, t_1, t_2, $\ldots$	A specific point in time. Subscripts will be used to distinguish between different points in time or to link a point in time to a certain event (e.g. reception of a message).
$p_{N_i}(t)$ or $p_i(t)$	The position (Cartesian coordinates) of a node N_i at time t.
$v_{N_i}(t)$ or $v_i(t)$	The velocity vector of a node N_i at time t.
$\\|\cdot\\|$	The Euclidean norm of a vector. Commonly used norms in this thesis are the distance $\\|p_2 - p_1\\|$ between two positions p_1 and p_2 and the speed $\\|v_i\\|$ of a node i.
m_{N_i} or m_i	The message broadcasted by N_i at time t in order to share its spatial state with other nodes. It is a quadruple containing the node's identity i, the transmission time t, the position $p_i(t)$, and the velocity $v_i(t)$.
f_0	The center frequency of the communication medium. We assume that nodes regularly tune their transmitters to this frequency when broadcasting messages.
c	The signal propagation speed. It depends on the communication medium used to exchange information. Typical values are the speed of light ($c \approx 299792458$ meters per second) for radio communication and the speed of sound ($c \approx 340$ meters per second) for ultrasound communication.
$\delta, \delta_1, \delta_2, \ldots$	The propagation delay of a transmission. Indices are used to refer to transmissions. We will use the same index as for transmission times in order to refer to a specific transmission.
$\Delta_{i,j}$	The intertransmission time of two signals transmitted at t_i and t_j, i.e., $\Delta_{i,j} = t_i - t_j$.
$\Delta'_{i,j}$, $\Delta^R_{i,j}$	The interarrival time of two signals originally transmitted at t_i and t_j. If there are multiple receivers, we replace the prime symbol with the respective receiver's identity.
$\rho_R, \rho_{R_1}, \ldots$	The Doppler shift experienced by the receiver indicated by the subscript. It is the expected difference between the transmit and receive frequency, i.e., $\rho_R = f_0 - f_R$.

Table 2.1: Symbols and Notations Overview

whether a *prover* P is in a region `R` of interest". While existing approaches are often summarized under the term "location verification protocols", there are different variants of the problem:

- *Distance bounding* [30] is the oldest approach and often used as a fundamental building block for other location verification protocols. Distance bounding techniques are used to prove that a node is in close physical proximity to another node. It provides methods to derive secure upper bounds of distances between nodes, i.e., `R` is a circle with a radius equal to the distance bound.

- *In-region verification* [29,39,40] protocols aim at verifying whether a node is located in a certain area. Such an area could be a room, a house, or even a stadium. As a matter of fact, Sastry et al. refer in their definition above to the subproblem in which `R` can have any arbitrary shape. As a consequence, distance bounding can be considered a special case of in-region verification.

- *Location verification* [31–38] is the most powerful of these concepts. Its goal is to verify the exact location claimed by an untrusted node. Both distance bounding and in-region verification can be substituted by a location verification protocol, but not vice versa. In actual location verification, `R` is reduced to a single point rather than an area.

All location verification[1] approaches follow a general pattern in which location-dependent and tamper-proof physical signal propagation characteristics are measured and compared to the claimed location. If the measurements do not comply with the claimed distance, location, or region, the information will be rejected by the system. It is worth mentioning that this approach is closely related to (secure) localization which has the goal to determine a node's exact location. In fact, location verification and localization techniques are based on the same physical signal properties. However, there is a slight difference which, in general, enables location verification protocols to accomplish their goal with less resources than localization systems. In addition, the increasing adoption of positioning systems such as GPS raises the need for complementary methods to provide security to critical applications and in hostile environments in an add-on fashion (compare subsection 2.1.1).

In the remainder of this section, we will first review existing location verification approaches and then discuss their applicability to mobile scenarios. We will highlight

---

[1] As common in the literature, we will collectively call the three subproblems *location verification* and

the challenges arising from mobility in networks and outline several open problems which will be tackled in this thesis.

2.2.1 Existing Methods

Since Brands and Chaum first addressed the problem of distance bounding in 1993 [30] and Sastry et al. later defined location (or in-region) verification in 2003 [29], many solutions and methods have been proposed in the literature to solve this problem. Based on the physical property used to verify locations, existing solutions can be broadly classified into methods using ranging [29–33], time-difference of arrival [34–36], hybrid [37, 38], and angle of arrival [39, 40] techniques to verify distances, regions, or locations.

Time of Flight (Ranging)

Ranging techniques are a popular basis for location verification schemes since they do not require time synchronization between the parties. In wireless networks, ranging techniques measure the Time of Flight (TOF) of a signal to estimate the distance between one or more verifiers and a prover. We distinguish between one-way and two-way ranging (better known as *distance bounding*). While two-way ranging has gained a lot of popularity within the research community, one-way ranging is far less common in the location verification context.

Čapkun et al. proposed a scheme in [34] where a challenge-response protocol is initiated by a mobile verifier from a known position. Before the response is transmitted, the verifier moves to an unknown position. The response is sent by the prover simultaneously via ultrasound and RF so that the verifier can estimate its distance to the prover based on the time-difference of arrival of the two signals due to their differing propagation speed (ranging). The security of the approach comes from the fact that although dishonest provers could modify the transmission times of the two response signals, they would need to correctly guess the verifier's new location in order to mimic the expected time difference of the two signals at the receiver. To the best of our knowledge, this is the only location verification protocol which uses a pure one-way ranging technique.

In contrast, two-way ranging protocols rely on a mix cryptographic techniques and time measurements to enable a verifier to establish an upper bound on the physical distance to a prover. The basic procedure is that a verifier sends cryptographic challenges to a prover which replies immediately with a response calculated from the challenge.

---

make a distinction only where necessary

By measuring the time between the transmission of the challenge and the reception of the response, the verifier obtains an upper bound for the processing and propagation delay. By using efficient response functions and sophisticated hardware to keep the ratio between processing and propagation delay as small as possible, the measured round-trip time provides a good *upper* bound estimator for the distance between sender and receiver. The security of these schemes is based on the assumption that a dishonest node cannot imitate a smaller distance since the signal cannot travel faster than c.

More formally, let $V \in \mathcal{V}$ and $P \in \mathcal{P}$ be a verifier and a prover. Let further $\hat{\delta}$ denote the round-trip time measured by V during a distance bounding challenge-response run. Then

$$\hat{\delta} = 2 \cdot \frac{\|p_P(t) - p_V(t)\|}{c} + \delta_P \ ,$$

where δ_P is the processing delay, that is, the time needed by P to send the response back to V. If $\delta_P \approx 0$, $\hat{\delta}$ becomes an estimator for twice the distance between V and P. Since the propagation delay is a physical constant in this equation, the prover cannot claim a location closer than $\|p_P(t) - p_V(t)\|$. To do so, δ_P would have to be negative which means that P would have to predict the challenge and send the response prior to the arrival of the challenge. This, however, is prevented by making the challenge unpredictable by cryptographic means.

Brands and Chaum were the first to introduce distance bounding in 1994 [30]. Sastry et al. then proposed using distance bounding with several verifiers for in-region verification in 2003 [29]. Later, concepts for using distance bounding for secure location verification were proposed by Singelee and Preneel [31] and Čapkun and Hubaux [32]. In both approaches, provers have to run a distance bounding protocol with three or more verifiers, resulting in three or more circles which intersect at the claimed location only if the claim was accurate. In the latest work, proposed in [33] by Perazzo et al., drones are used to perform distance bounding with a prover consecutively from several different locations. The locations are carefully chosen such that they form a triangle containing the prover's location. In this way an adversary claiming a false location inside the triangle needs to mimic a shorter distance to at least one of the locations chosen by the drone. As shown earlier by Čapkun et al. in [32], this is infeasible and the scheme is secure.

Time-difference of Arrival

Another measure for verifying locations is the Time Difference of Arrival (TDOA) of a signal at different locations. The TDOA is best-known from hyperbolic localization (also known as trilateration or multilateration) and works as follows. Let $V_1, V_2 \in \mathcal{V}$ be two *tightly synchronized* and *stationary* verifiers. A signal sent by prover P at time t is received by the verifiers at

$$t_1 = t + \|p_1 - p_P(t)\|/c \quad \text{and} \quad t_2 = t + \|p_2 - p_P(t)\|/c \; .$$

A central processing unit can collect the timestamps and calculate the TDOA

$$\begin{aligned} t_1 - t_2 &= \hat{t}_1 - \hat{t}_2 \\ &= (t + \|p_1 - p_P(t)\|/c) - (t + \|p_2 - p_P(t)\|/c) \\ &= \frac{\|p_1 - p_P(t)\| - \|p_2 - p_P(t)\|}{c} \; . \end{aligned}$$

Since the verifier locations and c are known, the only unknown in this equation is $p_P(t)$. Having at least three verifiers in a two-dimensional or four verifiers in a three-dimensional setting provides the central processing unit with enough such equations to find $p_P(t)$. More specifically, those locations which satisfy the TDOA equation above form a hyperbola in two dimensions or a hyperboloid in three dimensions. Each additional verifier results in another independent hyperbola which all cross $p_P(t)$. Hence, finding the intersection of the resulting hyperbolas (or hyperboloids) yields $p_P(t)$.

Although TDOA rather counts to the class of *localization* methods, it has been proposed by several researchers for location verification. The main advantage of TDOA over distance bounding is that it does not require the verifiers to communicate with the prover. Especially in the security context, this passiveness offers great advantages. In [34], for example, Čapkun et al. propose to measure the TDOA using *covert* base stations. Since the base stations do not have to transmit any signals to determine the TDOA, they can remain stealthy (i.e., their exact location is unknown to the attacker) to provide better robustness against signal timing attacks. Strohmeier et al. propose in [35] an “incomplete” multilateration scheme for verifying air traffic signals. They continuously compare single TDOA measurements with expected values. Although this approach cannot provide hard guarantees for detecting attacks, unsophisticated attackers can still be detected with only two verifiers. Achieving the same improvement but with stronger guarantees, Baker and Martinovic recently proposed another TDOA-based scheme in [36]. Their scheme relies on two verifiers, one fixed and the other one moving, to measure the TDOA of multiple location broadcasts by the prover. Since one verifier is changing its location between each of the prover’s transmissions, different TDOAs are expected each time. Analogously to traditional multilateration, each TDOA measurement reduces the set of possible locations of the transmitter to one arm of a hyperbola. By repeating the measurements at least three times (in two dimensions) and comparing the expected to the measured TDOAs, the adversary can be localized by intersecting the resulting hyperbolas.

Angle of Arrival

While most authors concentrated on approaches based on time measurements (TDOA and ranging), two works have been published which proposed using direction information of a received signal to perform in-region verification. One was published by Hu and Evans in [39]. They suggest using sectorized antennas to establish neighborhood relationships between nodes to prevent wormhole attacks. The key idea is to compare whether the relative direction of a signal, i.e., its Angle of Arrival (AOA) conforms with the expected direction. Similarly, Vora and Nesterenko suggested placing several verifiers with directional antennas around the region R of interest, whereas the antennas are directed towards the center of the region [40]. If a prover is located within R, the verifiers should all receive its signals. If a prover is located outside of R, there must be at least one verifier which does not face the prover and consequently does not hear its transmissions.

These approaches do not require any time synchronization, nor do angle of arrival measurements in general require any active transmissions by the verifiers. However, direction information is usually rather imprecise and noisy compared to timing information, especially in multipath environments. As a consequence, AOA measurements are less suited for accurate location verification and, to the best of our knowledge, no accurate location verification scheme based on the AOA has yet been published.

2.2.2 Challenges in Mobile Networks

An overview and comparison of all approaches is provided in Table 2.2. The table illustrates that the different verification schemes are all trade-offs between several system requirements and security. On the one hand, the number of verifiers can be reduced and time synchronization becomes dispensable if a system has the freedom to use active verification protocols to perform distance bounding. On the other hand, if financial, physical, and environmental conditions allow larger numbers of tightly synchronized verifiers, passive verification becomes feasible by using TDOA-based approaches. Other

Year	Publication	Method	Accuracy	Protocol	$\lvert\mathcal{V}\rvert$	Sync.
1994	Brands & Chaum [30]	Ranging	Distance	Active	1	No
2003	Sastry et al. [29]	Ranging	Region	Active	> 1	No
2004	Hu & Evans [39]	AOA	Region	Active	> 2	No
2004	Čapkun & Hubaux [37]	Ranging+TDOA	Location	Active	> 3	Yes
2005	Čapkun & Hubaux [32]	Ranging	Location	Active	> 3	No
2005	Singelee & Preneel [31]	Ranging	Location	Active	> 3	No
2006	Čapkun et al. [34]	Ranging	Location	Passive	1	No
2006	Vora & Nesterenko [40]	AOA	Region	Passive	> 4	No
2009	Chiang et al. [38]	Ranging+TDOA	Location	Active	> 3	Yes
2015	Perazzo et al. [33]	Ranging	Location	Active	1	No
2015	Strohmeier et al. [35]	TDOA	Location	Passive	> 1	Yes
2016	Baker & Martinovic [36]	TDOA	Location	Passive	2	Yes

Table 2.2: Comparison of existing location verification methods in chronological order.

approaches such as [34] and [35] trade security for lower system requirements by loosening the security guarantees or limiting the attacker's knowledge.

However, none of the aforementioned approaches can provide provable security, accuracy, and low system requirements in terms of required number of nodes, time synchronization, and communication overhead at the same time. This significantly limits their applicability to transportation systems where decades-long certification procedures and deployment cycles meet large numbers of vehicles and a permanent pressure to reduce costs. This combination prevents the use of both active and expensive passive verification methods although transportation systems are critical infrastructure and have a clear need for strong security guarantees.

Another factor which renders most approaches unusable in mobile networks is that they are not designed for mobile scenarios. The term *location* verification already suggests that the above approaches consider provers which remain more or less stationary during the verification procedure. As a result, their applicability to mobile provers is limited if not impossible since a node in motion is theoretically just an infinitesimal amount of time at the location to be verified. A more illustrative example is an en route aircraft moving at about 220 m/s which cannot come to a quick stop or reduce velocity to perform distance bounding.

We conclude that existing verification techniques do not match the needs of real-world mobile networks such as transportation systems. There is a lack of schemes which are specifically designed to deal with mobile nodes. At the same time, schemes must be passive, cheap, and provide a high level of security.

2.3 Effects of Mobility on Wireless Transmissions

This section summarizes the effects of node movements during and between transmissions on the wireless communication medium. We limit our considerations to effects which are solely caused by position changes and not by the environment or a combination of both. This explicitly excludes effects such as changing multipath propagation or interrupted line of sights due to obstacles. Even though these effects also offer opportunities for security due to their randomness and unpredictability, we rather seek provable security and therefore concentrate on well-defined effects. We specifically look at measurable effects caused by position changes during or between transmissions in the signal's time, frequency, and space domains.

2.3.1 Time Domain

Given an unsynchronized sender $S \in \mathcal{S}$, an unsynchronized receiver $R \in \mathcal{R}$, and two transmissions from S to R at times t_1 and t_2. As we specifically consider the effects of mobility, we assume that either S or R or both change their locations between the transmissions, i.e.,

$$p_S(t_1) \neq p_S(t_2) \;\vee\; p_R(t_1) \neq p_R(t_2)$$

holds. Our system's response variable for measuring effects in the time domain are the nodes' unsynchronized timestamps. From the perspectives of S and R, the transmissions results in two measurable local events, the transmissions at S and the signal arrivals at R. As a result, the sender S obtains timestamps $\hat{t}_{1,S}$ and $\hat{t}_{2,S}$ for the two signal transmissions and R obtains $\hat{t}_{1,R}$ and $\hat{t}_{2,R}$ for the two signal arrivals.

Let Δ_t be the *unknown* constant offset between the sender's and the receiver's clocks. The relationship between the transmission and reception timestamps can be modeled by[2]

$$\hat{t}_{i,R} = \hat{t}_{i,S} + \frac{\|p_R(t_i) - p_S(t_i)\|}{c} + \Delta_t$$

with $i \in \{1, 2\}$. A common approach to eliminating unknown offsets is considering differences or, in this case, periods instead of absolute timestamps. This is essentially the same principle as in the stopwatch example in subsection 2.1.3. While absolute values of different stopwatches are not comparable, the differences between values of the

---

[2] As mentioned above, we assume that c is much higher than the node's speed and approximate the receiver's location at the time of arrival by that at the time of transmission of the signal to avoid recursive definitions.

same stopwatch provide a globally comparable measure of time periods. The periods of interest in our case are the intertransmission and interarrival times of the two signals. Let $\Delta_{1,2}$ be the intertransmission time of two transmissions from S to R at times t_1 and t_2, i.e,

$$\Delta_{1,2} := t_2 - t_1 = \hat{t}_{2,S} - \hat{t}_{1,S} \ .$$

Let further $\Delta'_{1,2}$ be the interarrival time, i.e.,

$$\Delta'_{1,2} := \hat{t}_{2,R} - \hat{t}_{1,R} \ ,$$

and δ_i ($i \in \{1, 2\}$) the propagation delay of the i-th transmission, i.e.,

$$\delta_i := \frac{\|p_R(t_i) - p_S(t_i)\|}{c} \ .$$

Given the above relationship between the transmission and arrival timestamps, we obtain

$$\begin{aligned} \Delta'_{1,2} &= \hat{t}_{2,R} - \hat{t}_{1,R} \\ &= \left(\hat{t}_{2,S} + \frac{\|p_R(t_2) - p_S(t_2)\|}{c} + \Delta_t\right) - \left(\hat{t}_{1,S} + \frac{\|p_R(t_1) - p_S(t_1)\|}{c} + \Delta_t\right) \\ &= (\hat{t}_{2,S} - \hat{t}_{1,S}) - \left(\frac{\|p_R(t_2) - p_S(t_2)\|}{c} - \frac{\|p_R(t_1) - p_S(t_1)\|}{c}\right) \\ &= \Delta_{1,2} + (\delta_2 - \delta_1) \ . \end{aligned} \tag{2.1}$$

In words, the interarrival time measured by R differs from the measured intertransmission time measured by S exactly by the change in propagation delay between the two transmissions. Since propagation delay is a direct measure of the distance between sender and receiver, Equation 2.1 provides a direct method to measure distance changes without time synchronization:

$$\|p_R(t_2) - p_S(t_2)\| - \|p_R(t_1) - p_S(t_1)\| = ((\hat{t}_{2,R} - \hat{t}_{1,R}) - (\hat{t}_{2,S} - \hat{t}_{1,S})) \cdot c \ .$$

It is essential to note here that this measure, known as the Mobility-differentiated Time of Arrival (MDTOA) [41], is only available in mobile systems. If sender and receiver are both stationary, Equation 2.1 does not contain any information as both sides simply add up to the same value. Figure 2.1 illustrates the Mobility-differentiated Time of Arrival (MDTOA) with a simple example.

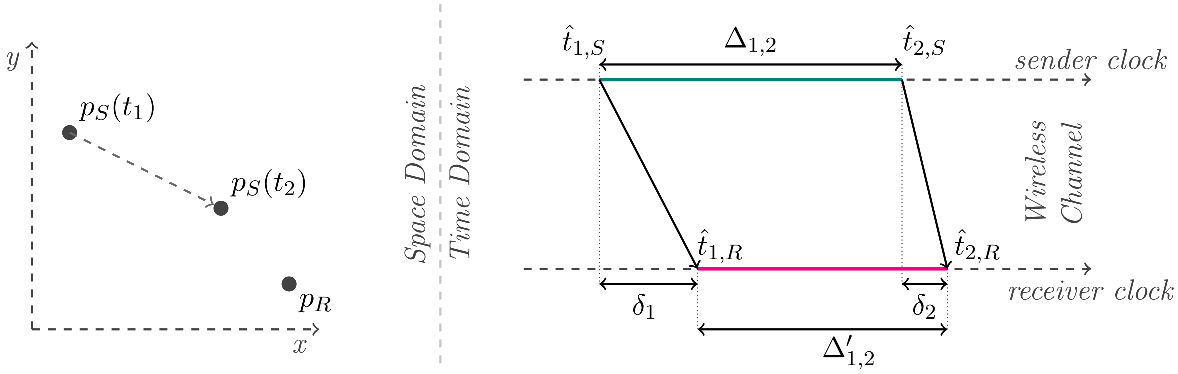


Figure 2.1: The effect of mobility on the interarrival times of two transmissions at receiver R. The interarrival time $\Delta'_{1,2}$ differs from the intertransmission time $\Delta_{1,2}$ exactly by the difference of the propagation delays δ_1 and δ_2 of the two transmissions. In this example, sender S moved closer to stationary receiver R between the two transmissions resulting in a negative difference $\delta_2 - \delta_1$ and thus in a interarrival time shorter than the intertransmission time.

We conclude that mobility can be detected by nodes in the time domain indirectly by measuring changes in the distance between sender and receiver. Moreover, this effect can be measured using the MDTOA without the need for time synchronization between the nodes. Sender and receiver simply have to share their node-local timestamps.

2.3.2 Frequency Domain

The effect of movement on a signal in the frequency domain is called the Doppler effect or Doppler shift. It is the change in wavelength[3] of a wave observed by a receiver R moving relative to the sender S of a signal. Let $v_{S/R}(t)$ be the relative movement of S towards R during a transmission at time t (compare left side of Figure 2.2). The velocity vector $v_{S/R}(t)$ is that part of $v_S(t)$ which defines the change rate of the distance between S and R. Its direction component is always pointing straight towards or straight away from R. Its speed component $\|v_{S/R}(t)\|$, called the *radial speed*, is the distance change rate between S and R at time t. It is worth noting that Doppler shift can be caused equally by movement of the sender during transmission, movement of a receiver during reception, or by both moving while transmitting and receiving a signal. For the sake of simplicity and with respect to our use of the Doppler effect later in section 4.3, we assume for the following definitions that S is mobile while R is stationary. An illustrating example scenario is provided in Figure 2.2 (left).

---

[3] Wavelength and frequency have a simple inversely proportional relationship: if one increases, the

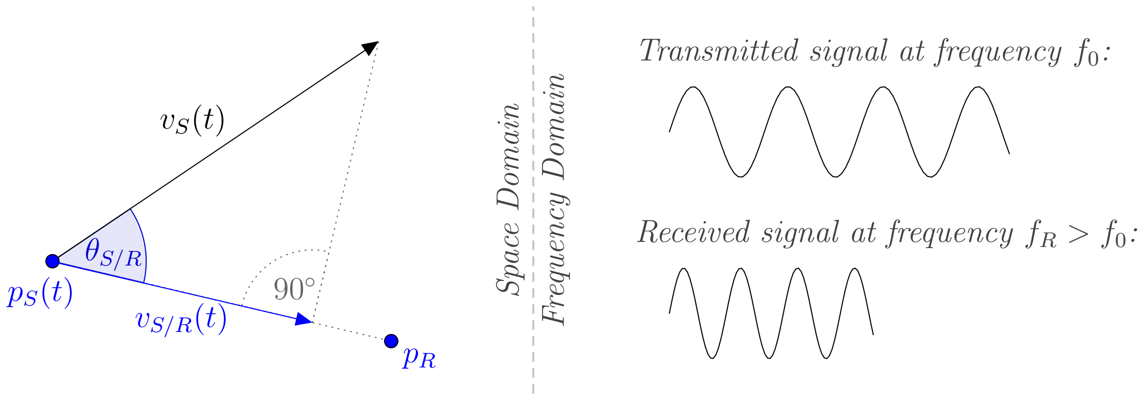

$v_S(t)$
$\theta_{S/R}$
$p_S(t)$
$v_{S/R}(t)$
90°
p_R
Space Domain
Frequency Domain
Transmitted signal at frequency f_0:
Received signal at frequency $f_R > f_0$:

Figure 2.2: At the time of transmission, the sender is located at $p_S(t)$ and moving with a velocity of $v_S(t)$. The radial velocity towards a stationary receiver R located at p_R is $v_{S/R}(t)$ and the radial speed, i.e., the rate of distance change is $\|v_{S/R}(t)\|$. In this example, S is moving towards R which results in a receive frequency f_R higher than the transmit frequency f_0.

Let $\cdot$ in a vector context be the dot product of the two vectors. Then,

$$\theta_{S/R}(t) = \arccos\left(\frac{v_S(t)\cdot(p_R(t)-p_S(t))}{\|v_S(t)\|\cdot\|p_R(t)-p_S(t)\|}\right)$$

is the angle between the sender's velocity vector $v_S(t)$ and the vector starting at $p_S(t)$ and ending at $p_R(t)$ at time t. The radial speed can be calculated by

$$\|v_{S/R}(t)\| = \cos\left(\theta_{S/R}(t)\right)\cdot\|v_S(t)\| = \frac{v_S(t)\cdot(p_R(t)-p_S(t))}{\|p_R(t)-p_S(t)\|}\ . \tag{2.2}$$

Noteworthy special cases for radial speeds are

- $\|v_{S/R}(t)\| = \|v_S(t)\|$ if S moves straight towards R ($\theta_{S/R}(t) = 0°$)
- $\|v_{S/R}(t)\| = -\|v_S(t)\|$ if S moves straight away from R ($\theta_{S/R}(t) = 180°$)
- $\|v_{S/R}(t)\| = 0$ if S moves on a circle around R ($\theta_{S/R}(t) = 90°$ or $\theta_{S/R}(t) = -90°$)

Finally, the frequency of S's signal observed by R becomes

$$f_R = \frac{f_0}{1-\frac{\|v_{S/R}(t)\|}{c}}$$

---

other one decreases at a fixed rate depending on the constant propagation speed c of the signal.

and the Doppler frequency shift (or simply Doppler shift) is

$$\rho_R = f_0 - f_R = \frac{f_0}{1 - \frac{c}{\|v_{S/R}(t)\|}} \ . \tag{2.3}$$

Given that c is often close to the speed of light since most real-world networks use radio communication, small Doppler shifts are to be anticipated due to the extremely small ratio between node speed and signal propagation speed. As we will show later, modern hardware is able to measure Doppler shifts in air traffic control communication since aircraft are moving at high speeds. However, measuring frequency shifts in radio communication caused by the movement of a pedestrian or a car is rather difficult. In such a case, ultrasound communication is preferred due to the much slower signal propagation speed (factor 10^{-6}).

Analogously to timestamps in the time domain, measuring the Doppler shift of a single transmission requires a tight frequency synchronization between sender and receiver. As before, such a synchronization requires either additional infrastructure or additional communication to establish and maintain synchronization. In unsynchronized networks, radio front-ends of receivers usually exhibit slight frequency offsets. In our case, this yields frequency measurements

$$\hat{f}_R = \hat{f}_0 + \rho_R + \Delta_f \ ,$$

where $\hat{f}_0$ is the transmission frequency that was actually used by S and Δ_f is the difference of S's and R's unknown frequency offsets. The alert reader might have noticed that ρ_R also contains the transmission frequency and should therefore be denoted by $\hat{\rho}_R$. However, we argue that frequency offsets of real-world radio front-ends are usually limited by the respective specifications and certification procedures. As a consequence, we consider the error introduced by the approximation $\hat{\rho}_R \approx \rho_R$ negligible. We show later in this thesis that this approximation is accurate enough for our purposes.

As in the previous section, this suggests using the difference of two measurements taken by the same node. In contrast to the MDTOA, however, the resulting "mobility-differentiated frequency of arrival" would be a measure of change in direction which is much less interesting since direction changes occur rather rarely in transportation systems. Nevertheless, the frequency differences of a signal received by multiple *synchronized* receivers, the so called Frequency Difference of Arrival (FDOA), provides indeed an useful indicator for a nodes' spatial state. Given two synchronized receivers

R_1 and R_2 and a unsynchronized sender S. Then both receivers have the same unknown frequency offset Δ_f relative to R and the Frequency Difference of Arrival (FDOA) is

$$\begin{aligned} \hat{f}_{R_1} - \hat{f}_{R_2} &= (\hat{f}_0 + \rho_{R_1} + \Delta_f) - (\hat{f}_0 + \rho_{R_2} + \Delta_f) \\ &= \rho_{R_1} - \rho_{R_2} \ . \end{aligned} \tag{2.4}$$

Through the radial speed $\|v_{S/R_i}(t)\|$ in ρ_{R_i}, the FDOA provides a measurable indicator which combines both the locations and velocities of all involved nodes. Even though it requires a frequency synchronization between receivers, the usually more difficult synchronization between receivers and transmitters is not required. This is a good match for transportation systems in which the receiving infrastructure may be synchronized more easily than moving, oftentimes resource-limited vehicles. Moreover, state of the art receiving hardware provides extremely high frequency stability. In fact, a recent study found that even extremely cheap (25 USD) software-defined radios provide high frequency stability with absolute offsets of less than 1 ppm [42]. With such a high stability, only an initial synchronization and occasional re-synchronization would be necessary to maintain the required frequency synchronization.

2.3.3 Space Domain

Communication in transportation systems usually happens somewhere on the VHF (AIS channels are at about 162 MHz) and the UHF (ADS-B uses either 978 MHz or 1.09 GHz) bands. An important property of these high frequencies is that radio waves cannot travel over the horizon or through obstacles. As a consequence, exchanging location information in such networks requires a direct line of sight between sender and receiver. This limitation combined with the mobility of nodes results in a constantly changing network topology. When nodes are moving, the line of sight connections between nodes can be interrupted by all kinds of obstacles in the environment such as houses, trees, mountains, or even other nodes. Moreover, in large networks communicating over extremely long distances[4], the connections between nodes will be interrupted by the earth's curvature as soon as they are beyond each others radio horizon.

This dependency on the line of sight provides us a way to probe the physical environment, albeit in a very simplistic way. If a receiver observes signals sent from a certain location, it knows that there is no obstacle between its own and the remote location.

---

[4] By long distances we mean, for example, air-ground communication in aviation (ADS-B) where line-of-sight communication is possible over ranges of up to 700 km [17].

However, in contrast to the time and frequency domain, where we have well-defined, accurate, and continuous response variables to measure the mobility of nodes, this space domain feature has a much more random nature and is limited in its applicability. More specifically, we can only measure the absence of obstacles, not their presence. The reason for that is that there can be many unpredictable reasons for communication losses. Firstly, a real-world environment is usually not static. New obstacles such as buildings and trees appear and also disappear over time. Secondly, inappropriate placements of antennas on nodes can result in temporary communication losses. For example, if an airplane has only one antenna installed on its belly, its own body might obscure the antenna when the aircraft turns with a high roll angle. Lastly, there are effects in wireless networks not related to the line of sight which can also cause communication losses. For instance, having too many nodes in a network communicating on the same frequency in an uncoordinated manner, the interference level can become too high resulting in an overloaded communication channel. Or multipath signal propagation can cause destructive interference which results in fading and, in the worst case, signal cancellation.

In summary, node mobility results in a highly dynamic topology which is strongly affected by the physical environment. However, real-world environments are usually extremely complex and taking advantage of them would require extensive and accurate models and knowledge which is often not available. Moreover, the variety of effects on wireless transmissions make the sheer presence of a signal a less suitable feature for security applications since measurements are subject to high levels of uncertainty. Nevertheless, we will outline a very basic approach to improving security in mobile networks by using space domain features in section 4.4.

In the next chapter, we show how these effects of mobility can be used to improve location verification in terms of system requirements. Afterwards, in chapter 4, we will extend the problem of location verification to meet the challenges of mobile provers such as those seen in transportation systems (see subsection 2.2.2). We will show how the effects of mobility can be further exploited and provide lightweight yet secure methods to verify provers in motion.

Truth was the only daughter of Time.

— Leonardo Da Vinci in his notebooks

3

Improving Location Verification with Mobile Verifiers

Contents

3.1 Network & Threat Model **30**
3.2 Verification Procedure **31**
  3.2.1 Directed Mobility vs. Mobility of Opportunity 32
  3.2.2 Protocol Properties 33
3.3 Security Analysis **34**
  3.3.1 Directed Mobility 38
  3.3.2 Mobility of Opportunity 40
  3.3.3 Movement Patterns 46
  3.3.4 Lessons Learned 48
3.4 Discussion **49**
  3.4.1 Stronger Adversaries 50

As the previous chapter has shown, mobility provides new measurable features in location aware networks. In this section, we investigate how one of these new features, the MDTOA, can be exploited to improve location verification. We show that mobile provers can achieve secure and accurate location verification with significantly lower system requirements than existing schemes (compare Table 2.2 on page 21). More precisely, our scheme is able to verify exact locations passively with two mobile verifiers and without assuming limited attacker knowledge, time synchronization, or active ranging protocols.

3.1 Network & Threat Model

By definition, location verification seeks a secure verification of provers which claim *single locations*. A consequence of this objective is that provers have to remain at the claimed location, at least during the verification process. For this reason, we consider provers which are *stationary*. As our aim is to take advantage of mobility, we assume that verifiers are able to move while executing the verification. This yields the following refined problem statement:

> A set of *unsynchronized* and *mobile* verifiers $\mathcal{V}$ wish to verify whether a (stationary) prover P is at a claimed location p.

In accordance with our model of location awareness (subsection 2.1.3), P broadcasts location claims $m_i(t_i) = (\hat{t}_i, p)$[5] several times, i.e., $i = 1, \ldots, n$ with $n > 1$. We assume that verifiers $V_j \in \mathcal{V}$ are constantly aware of their locations, i.e., they have means of positioning. Since we are particularly interested in the benefits gained through the mobility of verifiers, we demand $p_j(t_{i-1}) \neq p_j(t_i)$ for two subsequent broadcasts. We further assume that verifiers are passive receivers and there is no other communication between prover and verifiers than the prover's location broadcasts.

It is worth noting that the transmission timestamp $\hat{t}_i$ in m_i is only required to determine the MDTOA as the difference between intertransmission and interarrival time. The intertransmission time, however, could be fixed on a constant Δ (i.e., $\Delta_{i-1,i} = t_i - t_{i-1} = \Delta$ for all $i = 1, \ldots, n$) known to all nodes in the network. In that case, the transmission timestamp $\hat{t}_i$ could be omitted, saving a few bytes and providing better compatibility with existing protocols. However, including the timestamps into the location claims provides more flexibility which, for instance, adds support for random medium access

---

[5] For ease of notation, we will just write m_i and assume the definition of the real (global) transmission time t_i implicitly.

protocols such as ALOHA. We therefore assume in the following that the timestamps are transmitted and just provide the idea of omitting it on a side note. All analyses and definitions based on the timestamps also apply to the “fixed Δ”-version of the network model.

Adversary

The goal is to provide means for securely verifying the accuracy of a prover’s location claim. We therefore consider all information provided by the prover untrustworthy. More specifically, we assume that a malicious prover (adversary) has full control over reported timestamps, the real transmission intervals, and the actual claimed location p. As p is the actual property of interest here, we assume that in case of an attack, the adversary A is located at a location different from the claimed location, that is, $p_A \neq p$.

A consequence of our suspicion is that we have to assume that any information and action of the adversary has the goal to conceal the fact that $p_A \neq p$. In particular, the adversary tries to find timestamps and transmission times such that the location claims appear genuine to the verifiers. Since we aim at strong, provable security, we do not limit the adversary’s knowledge. The adversary has perfect knowledge of the verifiers’ locations at any point in time and it can even predict the verifiers’ future locations. This assumption is an important difference to the scheme proposed by Čapkun et al. in [34]. While their approach also uses mobile verifiers, the security is based on the adversary’s lack of knowledge about the verifiers’ location at the time of verification. With this limitation, the adversary is forced to resort to simple guessing for choosing the respective transmission times.

As for the adversary’s further capabilities, we assume that it is stationary during the verification process ($p_A(t_i) = p_A$ for all $i \in \{1, \dots, n\}$) and all verifiers receive the same signal. Loosening these assumptions by facing mobile or multi-device adversaries will be discussed in section 3.4. Finally, we assume that the verifiers are not compromised and they have secure means to determine their locations. Consequently, locations $p_j(t_i)$ and timestamps $\hat{t}_{i,j}$ are accurate for all $i \in \{1, \dots, n\}$ and $V_j \in \mathcal{V}$.

3.2 Verification Procedure

As mentioned in the previous section, prover P has to broadcast its location and timestamps $n > 1$ times in order to claim a location. On reception of each claim m_i, each verifier $V_j \in \mathcal{V}$ ($j = 1, \dots, |\mathcal{V}|$) measures the time of arrival $\hat{t}_{i,j}$ using its unsynchronized local clock and records its location $p_j(t_i)$. From a global view, each of the prover's broadcasts results in discrete "snapshots" of timestamps and locations of all nodes. To verify a location, each verifier $V_j \in \mathcal{V}$ checks for each broadcast $2 \leq i \leq n$ whether the MDTOA-based local verification condition

$$\hat{t}_{i,j} - \hat{t}_{i-1,j} \quad \overset{?}{=} \quad \hat{t}_i - \hat{t}_{i-1} + \left(\frac{\|p_j(t_i) - p\| - \|p_j(t_{i-1}) - p\|}{c} \right)$$

is satisfied. More concisely, with $\hat{\Delta}^j_{i-1,i} = \hat{t}_{i,j} - \hat{t}_{i-1,j}$ denoting the interarrival time measured by V_j, $\hat{\Delta}_{i-1,i} = \hat{t}_i - \hat{t}_{i-1}$ being the claimed intertransmission time, and $\hat{\delta}^j_i$ being the estimated propagation delay based on the claimed location, the verification condition becomes

$$\hat{\Delta}^j_{i-1,i} \quad \overset{?}{=} \quad \hat{\Delta}_{i-1,i} + \left(\hat{\delta}^j_i - \hat{\delta}^j_{i-1} \right) \ . \tag{3.1}$$

If the condition is satisfied, the verifier remains silent. If, however, the two sides of the equation do not add up, the verifier raises an alarm. Globally, the verification procedure terminates successful, i.e., P's location is verified, after n transmissions without any alarm.

Two things are important to note here. First, by *local* verification condition we mean that each verifier can check this condition without any further communication. Equation 3.1 only contains values that are either determined by the verifier itself or broadcasted by the prover. Second, we isolated all untrustworthy values on the right-hand side of the equation, while the left-hand side only consists of the genuine interarrival time measured by the verifier. In other words, we compare whether the measured interarrival time (left-hand side) complies with the interarrival time that is to be expected according to the data provided by the prover (right-hand side).

3.2.1 Directed Mobility vs. Mobility of Opportunity

The adversary has full control over the transmission timings and the claimed locations p. Even though this provides some degrees of freedom to the adversary, it cannot control Equation 3.1 at will. The reason for this is that it has no control over the verifiers' movement patterns which directly affect the verification through the distance changes. In principle, there are two options for verifiers to move during an ongoing verification. One obvious behavior would be that they adapt their movement once they learn p, that is, after they received the first location claim. By choosing appropriate paths with respect to p, the verifiers can improve the security of the scheme by avoiding adverse corner cases or geometrical conditions that might open up loop holes for the adversary. We call this conscious choice of movement patterns *directed mobility*.

This approach, however, has a major practical disadvantage. It significantly limits the scheme's scalability since it prevents batch verification. In general, verifiers often can and should only adapt their movement to one verifier at a time to prevent Sybil attacks. As a result, multiple provers can only be verified sequentially which, depending on the the number of provers, may result in unwanted latencies. We therefore consider a second movement strategy where verifiers move independently from the claimed location p. This approach is better suited for networks with large numbers of nodes or networks that require batch verification, e.g., to meet real-time requirements. Another example where directed mobility may not be the approach of choice are networks where ordinary but trusted nodes act as verifiers of opportunity while actually pursuing a different task. Referring to the latter example, we call this approach *mobility of opportunity*.

3.2.2 Protocol Properties

In terms of communication overhead, we summarize that our scheme only requires provers to transmit their location claims n times. Thus, keeping n small results in lower communication overhead. If there is no alarm, which should be the normal case, there is no further communication by the verifiers necessary. In addition to the low communication overhead, our protocol does not require any continuous time synchronization as it operates solely on local timestamps. This further reduces communication overhead since time synchronization is usually associated with the exchange of synchronization signals or additional infrastructure (compare subsection 2.1.3).

As for computational overhead, each verifier has to check Equation 3.1 n times during each verification process. One check only includes the calculation of two Euclidean distances and several basic arithmetic operations. We therefore argue that our protocol has an extremely low computational overhead and the total number of operations that must be executed by each verifier per verification is in $\mathcal{O}(n)$.

We conclude that our protocol is light-weight in terms of communication and computation. To ensure that it is also useful and secure, two properties must be shown according to Sastry et al. [29]:

- *Completeness:* If P and all verifiers behave according to the protocol, and P is located at p, then no verifier $V \in \mathcal{V}$ raises an alarm.

- *Security:* If all $V \in \mathcal{V}$ behave according to the protocol and accept P's claim, i.e., none of them raises an alarm, then P (or a party colluding with P) has a physical presence at p.

By definition of the MDTOA (subsection 2.3.1), Equation 3.1 is satisfied if the location claim is accurate. Thus, the completeness of our protocol is obvious. However, to prove and analyze the security of the scheme, a more profound analysis is required. The following section analyses the security of the scheme using both directed mobility and mobility of opportunity.

3.3 Security Analysis

We conduct our security analysis by deriving the theoretical constraints an adversary has to meet in order to successfully spoof a location. For this purpose, we start with the basic case $|\mathcal{V}| = 1$ and increment the number of verifiers one by one. This procedure has the advantage that we additionally obtain insights on the minimum number of verifiers required to establish security. This information is important for practical and secure implementations of our scheme since varying connectivity and message loss might in practice limit the number of usable verifiers. Based on our theoretical findings, we then extend our analysis towards the more complex mobility of opportunity scheme using simulations.

While we conduct our analysis in a two-dimensional space to maintain comprehensibility, extending the results to three dimensions is straightforward.

Single Verifier

Let $\Delta_{i-1,i} = t_i - t_{i-1}$ denote the *real* intertransmission time used by an adversary A located at p_A. The interarrival time of the adversary's claims measured by a single verifier V_j can be rewritten

$$
\begin{aligned}
\hat{\Delta}^j_{i-1,i} &= \hat{t}_{i,j} - \hat{t}_{i-1,j} \\
&= t_{i,j} - t_{i-1,j} \\
&= \left(t_i + \frac{\|p_j(t_i) - p_A\|}{c} \right) - \left(t_{i-1} + \frac{\|p_j(t_{i-1}) - p_A\|}{c} \right) \\
&= \Delta_{i-1,i} + (\delta^j_i - \delta^j_{i-1}) \ .
\end{aligned}
$$

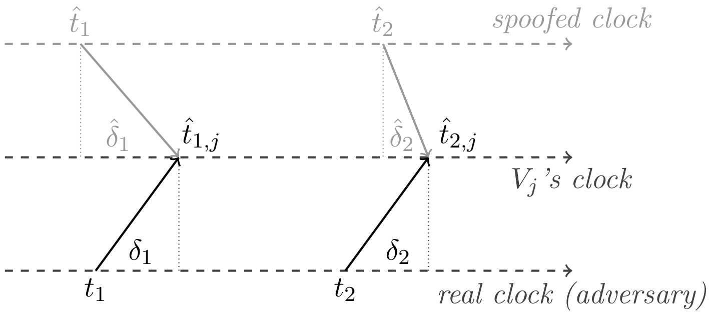


Figure 3.1: Example of an attack on a single verifier in the time domain. The adversary is adapting its transmission times such that the location claims have an interarrival time that is expected by the verifier.

By plugging this into the left-hand side of Equation 3.1 we obtain the following constraint. If the adversary wants to spoof a certain location p while actually being located at $p_A \neq p$, it needs to choose its intertransmission intervals such that

$$\Delta_{i-1,i} + (\delta_i^j - \delta_{i-1}^j) \quad = \quad \hat{\Delta}_{i-1,i} + (\hat{\delta}_i^j - \hat{\delta}_{i-1}^j)$$

holds. Considering only a single verifier, this can easily be achieved by simply choosing

$$\Delta_{i-1,i} = \hat{\Delta}_{i-1,i} + (\hat{\delta}_i^j - \hat{\delta}_{i-1}^j) - (\delta_i^j - \delta_{i-1}^j) \ . \tag{3.2}$$

In other words, the adversary can simply compensate the difference between the real propagation delays and those expected by V_j by choosing appropriate intertransmission intervals. As a result, our scheme is insecure for $|\mathcal{V}| = 1$ since adversaries can spoof arbitrary locations. Figure 3.1 illustrates such an attack in the time domain.

It is worth noting that if the adversary chooses its intertransmission intervals according to Equation 3.2, not only consecutive location claims appear genuine to V_j, but also each pair of the transitive hull of the interarrival times satisfies the verification condition: Let

$i_1, i_2 \in \{1, \dots n\}$ with $i_1 \neq i_2$. We assume $i_1 < i_2$ without loss of generality. Then the transitive intertransmission time can be summarized as

$$\begin{aligned}
\Delta_{i_1,i_2} &= \sum_{i_1 < i \leq i_2} \Delta_{i-1,i} \\
&= \sum_{i_1 < i \leq i_2} \left(\hat{\Delta}_{i-1,i} + (\hat{\delta}_i^j - \hat{\delta}_{i-1}^j) - (\delta_i^j - \delta_{i-1}^j) \right) \\
&= \sum_{i_1 < i \leq i_2} \left(\hat{\Delta}_{i-1,i} \right) + (\hat{\delta}_{i_2}^j - \hat{\delta}_{i_1}^j) - (\delta_{i_2}^j - \delta_{i_1}^j)
\end{aligned}$$

and V_j's interarrival time becomes

$$\begin{aligned}
\hat{\Delta}_{i_1,i_2}^j &= \Delta_{i_1,i_2} + (\delta_{i_2}^j - \delta_{i_2}^j) \\
&= \sum_{i_1 < i \leq i_2} \left(\hat{\Delta}_{i-1,i} \right) + (\hat{\delta}_{i_2}^j - \hat{\delta}_{i_1}^j) \ ,
\end{aligned}$$

which is exactly what V_j expects.

Two Verifiers

We continue our analysis by considering a system with two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$. Then, Equation 3.2 must be satisfied for both verifiers, i.e.,

$$\begin{aligned}
\Delta_{i-1,i} &= \hat{\Delta}_{i-1,i} + (\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}) - (\delta_i^{j_1} - \delta_{i-1}^{j_1}) \\
\Delta_{i-1,i} &= \hat{\Delta}_{i-1,i} + (\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}) - (\delta_i^{j_2} - \delta_{i-1}^{j_2})
\end{aligned}$$

must both hold for all $i = 2, \dots, n$. By equating and re-arranging both constraints, we can conclude that such a $\Delta_{i-1,i}$ exists if and only if the following requirement is met:

$$\begin{aligned}
(\delta_i^{j_1} - \delta_{i-1}^{j_1}) - (\delta_i^{j_2} - \delta_{i-1}^{j_2}) &= \left(\hat{\Delta}_{i-1,i} + (\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}) \right) - \left(\hat{\Delta}_{i-1,i} + (\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}) \right) \\
&= (\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}) - (\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}) \ . \qquad (3.3)
\end{aligned}$$

This means that the intertransmission interval $\Delta_{i-1,i}$ only exists if the adversary is either located at a position where the differences of distances[6] to each verifier changes between two transmissions exactly by the same amount as the differences of distances from p to V_{j_1} and V_{j_2}. Alternatively, since the adversary is clairvoyant, it can also try to find and

---

[6] Here, we interpret propagation delays as direct representatives of distances.

claim a location p (e.g., within an area of interest) which satisfies this constraint. From a mathematical point of view both strategies are equal since the adversary either tries to find a location p_a (left-hand side of Equation 3.3) which matches a given p (right-hand side) or vice versa.

From the verifier perspective, however, it makes more sense to analyze for a given p whether there is a location $p_A \neq p$ which also satisfies Equation 3.1 for all verifiers. Hence, without loss of generality, we further analyze the existence of such a location p_A given a claimed location p. Since p and the verifier's locations are fixed in that case, the only free parameter left in Equation 3.3 is p_A and we therefore summarize its right-hand side by a constant

$$k_i(p) = (\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}) - (\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2})$$

which yields the requirement

$$\delta_i^{j_1} - \delta_{i-1}^{j_1} = \delta_i^{j_2} - \delta_{i-1}^{j_2} + k_i(p) \tag{3.4}$$

for two consecutive transmissions of a false claim. As a result, p can only be spoofed from locations p_a at which the distance change between two transmissions from A to V_{j_1} differs exactly by $k_i(p)$ from that to V_{j_2}.

Geometrically, the set of coordinates with an equal absolute distance difference to two fixed points defines a hyperbola (or a hyperboloid in 3D). Accordingly, the left-hand side of Equation 3.4 defines one arm of a hyperbola with foci $p_{j_1}(t_i)$ and $p_{j_1}(t_{i-1})$ and distance difference $(\delta_i^{j_1} - \delta_{i-1}^{j_1}) \cdot c$ and the right-hand side defines one arm of a hyperbola with foci $p_{j_2}(t_i)$ and $p_{j_2}(t_{i-1})$ and distance difference $(\delta_i^{j_2} - \delta_{i-1}^{j_2} + k_i(p)) \cdot c$. The two sides of Equation 3.4 define only one arm of each hyperbola because we consider signed rather than absolute distance differences. We conclude that the set of locations p_A that satisfy Equation 3.4 is equal to the set of intersections of the two aforementioned hyperbola arms defined by either side of the equation.

In summary, by introducing a second verifier, the attacker's degrees of freedom are significantly reduced as it cannot be located at arbitrary positions anymore in order to spoof a certain location. Figure 3.2 shows an example scenario with two transmissions of location claims for p, two verifiers, and the implicit curve defined by Equation 3.4 (dashed line). A possible location p_A of an adversary is also indicated, although it could be anywhere on the dashed line. It is worth mentioning that p is by construction on the implicit curve. While this is natural since the legitimate p must satisfy the above

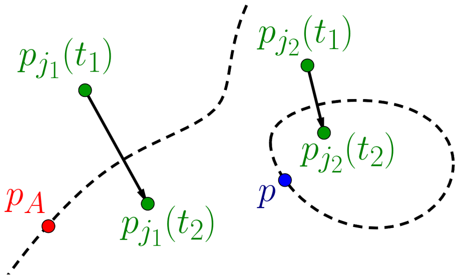


Figure 3.2: Example movement of two verifiers V_{j_1} and V_{j_2} between two transmission of a location claim for position p. The dashed curve is the resulting constraint according to Equation 3.4 for an adversary's location p_A. The adversary remains undetected only if it is located on the dashed curve.

constraint, it also implies that there are locations within the area of interest (e.g., nearby p) where an adversary could be located without being detected.

3.3.1 Directed Mobility

We did not distinguish between directed mobility and mobility of opportunity in our security analysis so far. As we mentioned above, being able to adapt the verifiers' movement patterns with respect to the claimed location can improve the security of our scheme. We will now demonstrate this statement by proposing a movement pattern for verifiers which establishes *provable* security with a minimum configuration of two transmissions and two verifiers.

Theorem 3.3.1 *If one verifier moves exactly towards* p *without passing it while another one moves exactly away from* p*, then our method is secure for* $n = 2$.

Proof: Given two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ and two location claim broadcasts m_1 and m_2 ($t_2 > t_1$ w.l.o.g.) sent by a prover P. Let us assume that the verifiers do not change their velocity between t_1 and t_2, i.e., $v_j(t) = v_j(t_1) = v_j$ for $t_1 \leq t \leq t_2$ and $j \in \{j_1, j_2\}$. We further assume that, without loss of generality, V_{j_1} is heading directly towards and V_{j_2} directly away from p without passing it. More formally, there is an $x_1 \in \mathbb{R}$ such that $x_1 > \Delta_{1,2} = (t_2 - t_1)$ and

$$p = p_{j_1}(t_1) + x_1 \cdot v_{j_1} \ .$$

Analogously, there is an $x_2 \in \mathbb{R}$ such that

$$p = p_{j_2}(t_1) - x_2 \cdot v_{j_2} \ .$$

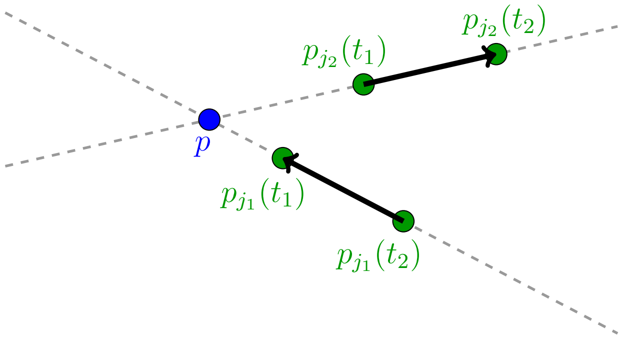


Figure 3.3: Verification scenario according to Theorem 3.3.1. V_{j_1} is moving towards p while V_{j_2} is moving away from p.

Note that the constraint $x_1 > \Delta_{1,2}$ is only needed for V_{j_1} to ensure that it does not pass p between t_1 and t_2. Since V_{j_2} is moving away from p, such a constraint is not necessary. An example scenario of a verification according to these assumptions is shown in Figure 3.3.

Then, the following holds:

$$\hat{\delta}_2^{j_1} = \hat{\delta}_1^{j_1} - \frac{\hat{\Delta}_{1,2} \cdot \|v_{j_1}\|}{c} \quad \text{and} \quad \hat{\delta}_2^{j_2} = \hat{\delta}_1^{j_2} + \frac{\hat{\Delta}_{1,2} \cdot \|v_{j_2}\|}{c} \ ,$$

since the distances between the two verifiers and p change exactly by the distance covered by the verifier between the two broadcasts. Plugging this into Equation 3.1 yields

$$\begin{aligned}
\hat{\Delta}_{1,2}^{j_1} &\overset{?}{=} \hat{\Delta}_{1,2} + \left(\hat{\delta}_2^{j_1} - \hat{\delta}_1^{j_1}\right) \\
&= \hat{\Delta}_{1,2} + \left(\hat{\delta}_1^{j_1} - \frac{\hat{\Delta}_{1,2} \cdot \|v_{j_1}\|}{c} - \hat{\delta}_1^{j_1}\right) \\
&= \hat{\Delta}_{1,2} - \hat{\Delta}_{1,2} \cdot \frac{\|v_{j_1}\|}{c}
\end{aligned}$$

and

$$\begin{aligned}
\hat{\Delta}_{1,2}^{j_2} &\overset{?}{=} \hat{\Delta}_{1,2} + \left(\hat{\delta}_2^{j_2} - \hat{\delta}_1^{j_2}\right) \\
&= \hat{\Delta}_{1,2} + \left(\hat{\delta}_1^{j_2} + \frac{\hat{\Delta}_{1,2} \cdot \|v_{j_2}\|}{c} - \hat{\delta}_1^{j_2}\right) \\
&= \hat{\Delta}_{1,2} + \hat{\Delta}_{1,2} \cdot \frac{\|v_{j_2}\|}{c} \ .
\end{aligned}$$

Analogously to Equation 3.3, an adversary A located at $p_A \neq p$ would have to choose an intertransmission interval $\Delta_{1,2}$ such that the equations

$$\begin{aligned} \Delta_{1,2} + (\delta_2^{j_1} - \delta_1^{j_1}) &= \hat{\Delta}_{1,2} - \hat{\Delta}_{1,2} \cdot \frac{\|v_{j_1}\|}{c} && (3.5) \\ \Delta_{1,2} + (\delta_2^{j_2} - \delta_1^{j_2}) &= \hat{\Delta}_{1,2} + \hat{\Delta}_{1,2} \cdot \frac{\|v_{j_2}\|}{c} && (3.6) \end{aligned}$$

are satisfied.

Let us now assume A is not located in line with p and V_{j_1}, or, more formally, there is no $x_1 \in \mathbb{R}$ such that $p_A = p_{j_1}(t_1) + x_1 \cdot v_{j_1}$. Then

$$(\delta_2^{j_1} - \delta_1^{j_1}) > -\hat{\Delta}_{1,2} \cdot \frac{\|v_{j_1}\|}{c}$$

since V_{j_1} does not move exactly towards A and thus, the distance change is not at its maximum negative amplitude (right-hand side) given the velocity of V_{j_1}. As a consequence, A has to choose an intertransmission time shorter than the claimed one, that is $\Delta_{1,2} < \hat{\Delta}_{1,2}$, in order to satisfy Equation 3.5. Then, however, Equation 3.6 cannot be satisfied since

$$(\delta_2^{j_2} - \delta_1^{j_2}) > \hat{\Delta}_{1,2} \cdot \frac{\|v_{j_2}\|}{c}$$

would be required. This means that the distance between V_{j_2} and A would have to change more than actually possible. Hence, A must be in line with p and V_{j_1} in order to satisfy both equations.

We can show analogously that A must also be in line with p and V_{j_2}. As a result, A must be located on two lines which cross p. Unless V_{j_1} and V_{j_2} are in line, these two lines are different and since two different lines can only have one intersection we conclude that p is the only location from which a sender can satisfy both equations at the same time. Thus, our protocol is secure and Theorem 3.3.1 holds.

□

3.3.2 Mobility of Opportunity

The movement pattern described in the previous section can only be applied if there is only one location to be verified at a time. For scenarios with more than one prover, a movement strategy independent from p is desired. So far, we considered only two consecutive transmissions of the location claim in the presence of two verifiers. Having more than two verifiers ($|\mathcal{V}| > 2$) or more than two transmissions ($n > 2$) each reduce the degree of freedom of the adversary by adding more implicit curves to the constraints. More specifically, since the verifiers move between each transmission, the focal points for the implicit curve defined by Equation 3.4 change for every $i \in \{2, \ldots, n\}$ and each pair $V_{j_1}, V_{j_2} \in \mathcal{V}$. As a result, A needs to be located at an intersection of $(n-1) \cdot \binom{|\mathcal{V}|}{2}$ different implicit curves in order to remain undetected when claiming $p \neq p_A$. Moreover, this set of intersections can be assumed to be finite since the curves are not periodic.

The number of these intersections can be considered a direct measure of the security of our scheme. The smaller this number, the less $p_A \neq p$ satisfy all constraints allowing an adversary to remain undetected. Our scheme is in particular secure if there is only one intersection of all curves (which is p by construction) since false claims would then violate Equation 3.1 for at least one verifier.

Most related problems are of a simple hyperbolic nature and can often be analyzed algebraically. Unfortunately, having more than one mobile node makes the exact analysis hard because each moving element contributes to the equations [43]. For example, in contrast to the analysis of intersections of a set of hyperbolas, which is common, e.g., for TDoA or ranging-based approaches, we face curves defined by intersections of intersections of hyperbolas with multiple parameters. As Figure 3.2 visually suggests, these curves are of a higher order than hyperbolas which makes an exact analysis of the intersections extremely difficult. Although there exist methods to decrease the computational complexity (e.g., homogeneous coordinates [44]), we could not find any analytical method to calculate the number of intersections in a general way. We therefore continue our analysis by extending our theoretical findings with simulations analyzing the behavior of the intersections with respect to the verifiers' movements.

Simulations

We implemented a simulation framework in MATLAB® which allows us to analyze the intersections for arbitrary constellations of verifiers and provers. By controlling the movements of verifiers between the reception of location claims, we show the effect of geometry on the security of our approach and identify beneficial movement strategies for verifiers.

Simulation Design: In accordance with our verification process, we implemented our simulations as a discrete-event simulation. The events are the transmissions of a location claim and we recorded the locations of all verifiers at each transmission. Based on the recorded locations and p we setup the nonlinear equation system consisting of all

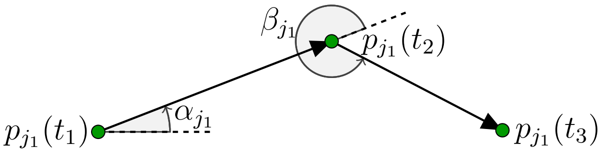


Figure 3.4: Description of a verifier V_{j_1}'s movement in our simulations. The initial direction is the counterclockwise angle α_{j_1} relative to a horizontal axis through $p_{j_1}(t_1)$. After the initial step (i.e. for $i > 1$), we only consider the direction changes (β_{j_1}, γ_{j_1}, ...), i.e., the counterclockwise angle between the old and the new direction.

$(n-1) \cdot \binom{|\mathcal{V}|}{2}$ instances of Equation 3.4. Then, we calculated all solutions to the system, i.e., the intersections of the curves, within a pre-defined area of interest using the solver `fsolve` of MATLAB®'s optimization toolbox. To find all different intersections, we used 50 equidistant points on one of the curves as initial points.

In order to analyze the effect of the verifiers' movements on the number of intersections within the area of interest, we define the movements as depicted in Figure 3.4. At the reception of the prover's first claim ($i = 1$), a verifier V_{j_1} is located at $p_{j_1}(t_1)$ and moves into direction α_{j_1} at a constant speed s, i.e., $\|v_{j_1}\| = s$. The direction of movement α_{j_1} is the counterclockwise angle between v_{j_1} and the x-axis. When the prover re-transmits the location claim at t_2, the verifier moved to location

$$p_{j_1}(t_2) = p_{j_1}(t_1) + (t_2 - t_1) \cdot v_{j_1} = p_{j_1}(t_1) + \Delta_{1,2} \cdot v_{j_1} \ .$$

For further transmissions ($i > 1$) we consider only the direction change β_a, γ_a, and so on. In summary, a verifier's movement during the verification process can be completely described by its initial location $p_{j_1}(t_1)$, its velocity v_{j_1}, the transmission times t_i, the initial direction α_{j_1}, and the direction changes β_{j_1}, γ_{j_1}, ... between the receptions. For simplicity, we used a constant intertransmission interval Δ, i.e., $\Delta_{i-1,i} = \Delta$ for all $1 < i \leq n$. Extending this to arbitrary intervals is straightforward.

Parameter Selection: To keep our simulations realistic, we have chosen the simulation parameters based on the following real-world examples. The speed of the verifiers is assumed to be in the range of off-the-shelf drones (10-30 m/s). The distance covered between two transmissions of a location claim is the product of this speed and the intertransmission interval Δ. This provides some flexibility in practice since while the speed is usually limited, Δ can be increased to obtain larger MDToA values. This is an important factor when considering real-world systems with noisy timestamps. Having larger values improves the noise ratio and thus the accuracy of the system. As a consequence, choosing Δ is a trade-off between the time needed to verify a location and accuracy. However, for simplicity, we set $\Delta = 1$ s for all our simulations. The area of interest considered in our simulations is motivated by the size of a football stadium and set to a rectangle of 209x255 m^2. All verifiers and location claims must be within this area.

As for the adversary's location, we allow it to be located outside this area but limit its distance to the verifiers in the following way. We assume that each verifier has a circular reception range with a radius sufficient to cover the largest possible distance between two locations within the area of interest, i.e., the area's diagonal. As a consequence, if all verifiers are located at the same side of the area, an attacker located outside the area could still be in their coverage. We therefore extend the area in which we search for intersections with a safety margin of the length of the diagonal of the area of interest.

An example scenario matching this parameter selection would be a location-based service which should only be available to people within the stadium. To access the service without having to pay entry, the adversary tries to spoof a location within the stadium while being located outside (but in range). Drones are hovering in the stadium and act as verifiers.

Results

We know from our formal analysis that the adversary's possible locations are reduced to a set of intersections $\mathcal{I}$ of implicit curves for $n > 2$ and $|\mathcal{V}| > 1$. These implicit curves are defined by the locations of the considered pair of verifiers and the claimed location (see Equation 3.4). While we cannot control the claimed location, the movements of the verifiers can be controlled and we are therefore interested in patterns which minimize $|\mathcal{I}|$, at best to $\mathcal{I} = \{p\}$. Moreover, since new curves are added with each additional verifier and re-transmission, we are also interested in the least required number of transmissions n and number of verifiers $|\mathcal{V}|$ which reduce the intersections to the claimed location p.

Speed s: The speed of the verifiers defines the distance covered by a verifier between the periodic re-transmissions of a location claim. To evaluate whether the resulting step width has an impact on the number of intersections, we generated 10k scenarios for the basic case $|\mathcal{V}| = 2$ and $n = 3$. Each verifier starts at a random location and moves into a random direction at different speeds $10 \leq s \leq 100$ m/s. For each scenario, we recorded

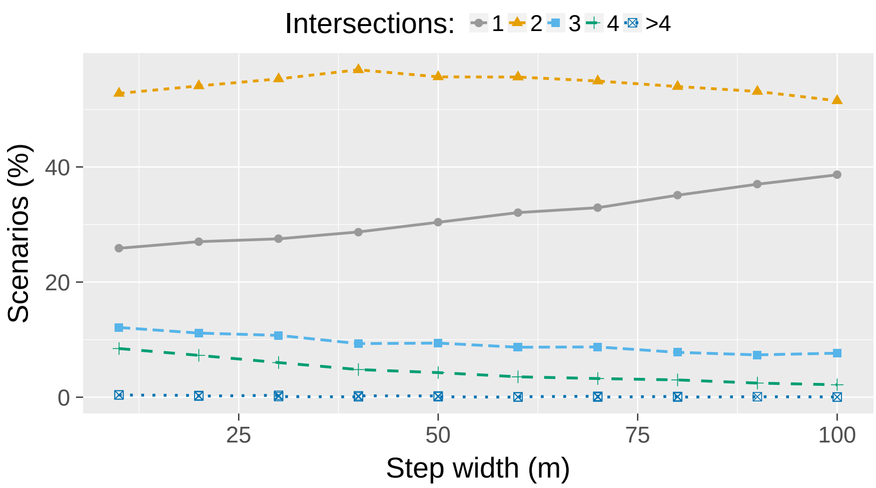


Figure 3.5: Effect of step width/speed on the number of intersections. The gray solid line is the percentage of the cases in which the location was securely verified. The other lines represent the percentages where an adversary could have been located at an increasing number of locations other than p.

the number of intersections $|\mathcal{I}|$. We did not consider larger speeds since they would be unrealistic given an area of interest of size 209x255 m^2.

The results are shown in Figure 3.5. While the percentage of scenarios in which the claimed location could be securely verified (gray solid line) slightly increased with increasing step width, the percentage of $|\mathcal{I}| = 2$ was constantly over 50%. For smaller v, there were even about 20% of scenarios in which an adversary could have chosen between two (dashed blue with squares) or three (green dashed with pluses) locations different to p which also satisfied Equation 3.1 for both verifiers.

We conclude that the step width (or speed) has only a minor effect on the number of intersections. On the one hand, this means that the step width does not provide much room for improving the security. On the other hand, however, this also means that slow verifiers do not suffer big disadvantages.

Number of transmissions n **and verifiers** $|\mathcal{V}|$**:** Both numbers n and $|\mathcal{V}|$ affect the security by controlling the number of curves whose intersections define $\mathcal{I}$. As mentioned above, the adversary's location p_A must lie on $(n-1) \cdot \binom{|\mathcal{V}|}{2}$ implicit curves in order to successfully spoof p. As before, we start our analysis with the smallest configuration ($n = 3$ and $|\mathcal{V}| = 2$) and generated 10k random verification scenarios with random initial verifier locations, random αs and βs, and random speeds $10 \leq s \leq 100$ m/s.

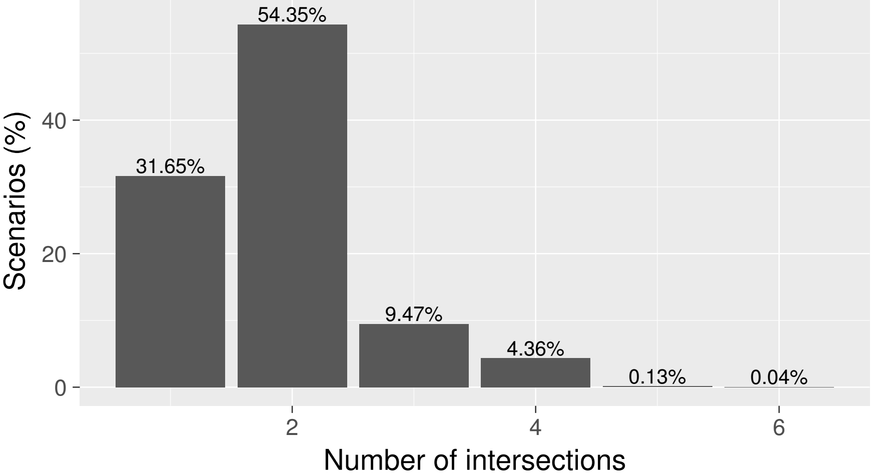


Figure 3.6: Distribution of the number of intersections over all 10,000 random simulation runs with 2 verifiers and 3 transmissions of the location claim.

The distribution on the number of intersections over all 10k simulation runs is shown in Figure 3.6. As expected, the results were equal to those for the average step width of 55 m shown in Figure 3.5. Only 31.65% of all tested scenarios could be securely verified with the basic configuration of $n = 3$ and $|\mathcal{V}| = 2$. A large number of scenarios resulted in two intersections (54.35%). The probability for more than two intersections, however, is significantly lower (less than 10% for three intersections). The highest number of intersections observed was 6. An example case with 6 intersections is shown in Figure 3.7.

While the case shown in Figure 3.7 nicely illustrates the curvy behavior of the two curves around p, it also indicates that the probability that a third curve of such an erratic nature would cross one of the intersections is extremely low. In fact, we conducted another 10k random simulations for $n = 4$ as well as for $|\mathcal{V}| = 3$ and the number of intersections dropped to 1 for all tested scenarios, meaning that *our verification scheme is secure for* $n > 3$ *and* $|\mathcal{V}| > 2$.

We can conclude that if the verifiers move completely uncontrolled (mobility of opportunity) within the area of interest, 31.65% of the location verification scenarios can be securely verified with $n = 2$ verifiers and $m = 3$ transmissions. In order to securely verify the other 68.35%, at least one additional transmission ($m > 3$) or at least one additional verifier ($n > 2$) is required. That means that if, for instance, the intertransmission interval is $\Delta = 1$ s, an adversary would be discovered after 2 s in 31.65% of the scenarios and at latest after 3 s, resulting in an average verification time of 2.6835 s. Conversely, with an additional verifier, the verification time is reduced to 2 s.

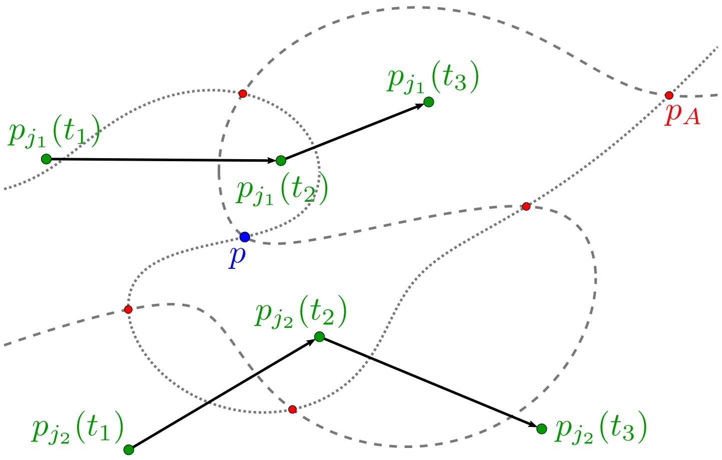


Figure 3.7: Example scenario where two verifiers V_{j_1} and V_{j_2} receive three transmissions of a location claim. The dashed curves are the resulting constraints according to Equation 3.4 for an adversary's location p_A. The adversary remains undetected only if it is located on one of the 6 intersections of the two curves.

3.3.3 Movement Patterns

The previous results show that our scheme is secure for $n > 3$ or $|\mathcal{V}| > 2$. However, depending on the use case, minimum verification time and minimum number of verifiers might be required. For example, if the area of interest is larger, parts of the area might only be covered by some verifiers. In addition, message loss might reduce the number of messages received by a sufficient number of verifiers. To further increase the efficiency of our scheme for a better robustness against such problems, we now analyze whether the security of the minimum configuration ($n = 3$ and $|\mathcal{V}| = 2$) improves if the verifiers' movements are controlled. Being able to securely verify a larger fraction of locations with the minimal configuration reduces the average verification time and the required number of verifiers. However, in contrast to directed mobility considered in subsection 3.3.1, we now concentrate on pre-defined movement patterns that are independent from the claimed location p.

Our next set of simulations aims at shedding light on the influence of the movement directions α and β on $|\mathcal{I}|$. It is worth noting that we do not analyze the effect of the initial location since we assume that the adversary controls the point in time when the verification process is initiated. Hence, the verifiers can only control what happens after the first location claim was received. In addition, we do not consider movement patterns as functions of p since this would prevent batch verification and potentially make the verification process vulnerable to Sybil attacks.

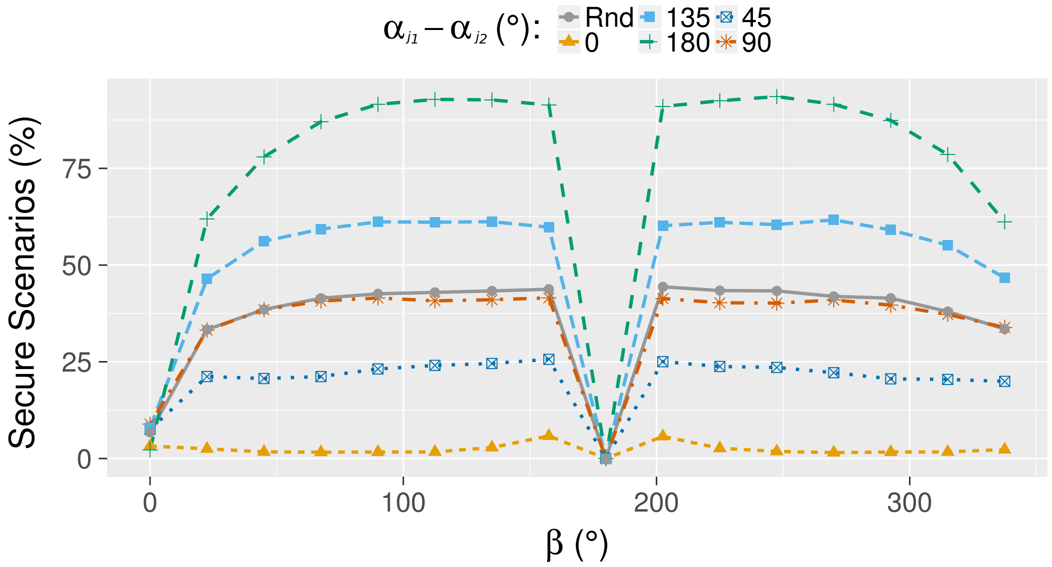


Figure 3.8: Effect of relative movement ($\alpha_{j_1} - \alpha_{j_2}$) and different β on the number of intersections. This graph only shows the percentage of the 10,000 random scenarios that were secure, i.e., the set of intersections is $\mathcal{I} = \{p\}$.

For all subsequent simulations we set the speed of the verifiers to that of commercial off-the-shelf drones such as DJI's Phantom 4, i.e., $v = 20$ m/s. The turns of the verifiers between the two steps are controlled by β_{j_1} and β_{j_2}. To keep our scheme light-weight, we assume that the verifiers do not communicate for coordination and assume constant pre-defined $\beta_{j_1} = \beta_{j_2} = \beta$. However, since the curves determining $|\mathcal{I}|$ not only depend on β but also on α_{j_1} and α_{j_2}, we further analyze how the difference between the two angles, i.e., the relative direction of the verifiers to each other affects the intersections. We again conducted 10k random simulations for different combinations of β and $\alpha_{j_1} - \alpha_{j_2}$.

The results are shown in Figure 3.8. The graph shows that both the effect of β and that of $\alpha_{j_1} - \alpha_{j_2}$ on $|\mathcal{I}|$ are almost independent from each other. Regardless of the difference in direction, any β close to 0° (respectively 360°) should be avoided. For large direction differences $\alpha_{j_1} - \alpha_{j_2}$, the best choice for β is around 110° or 250°. Note that both angles represent the same absolute change in direction since $360° - 250° = 110°$.

An interesting special case is $\beta = 180°$, i.e., the third location of each verifier is the same as the first one ($p_j(t_3) = p_j(t_1)$). As a result, the implicit curve generated by the first two transmissions coincides with that of the second and third transmission, resulting in infinite intersections. More specifically, given two verifiers V_{j_1} and V_{j_2} receiving three transmissions of a location claim for p. An adversary's location p_A must satisfy the following system of instances of Equation 3.4:

$$\begin{aligned}(\delta_2^{j_1} - \delta_1^{j_1}) &= (\delta_2^{j_2} - \delta_1^{j_2}) + k_2(p)\\ (\delta_3^{j_1} - \delta_2^{j_1}) &= (\delta_3^{j_2} - \delta_2^{j_2}) + k_3(p)\ .\end{aligned}$$

If $\beta = 180°$, i.e, $p_{j_1}(t_3) = p_{j_1}(t_1)$ and $p_{j_2}(t_3) = p_{j_2}(t_1)$ then

$$\begin{aligned}k_3(p) &= (\hat{\delta}_3^{j_1} - \hat{\delta}_2^{j_1}) - (\hat{\delta}_3^{j_2} - \hat{\delta}_2^{j_2})\\ &= (\hat{\delta}_1^{j_1} - \hat{\delta}_2^{j_1}) - (\hat{\delta}_1^{j_2} - \hat{\delta}_2^{j_2})\\ &= -k_2(p)\end{aligned}$$

and thus

$$\begin{aligned}& (\delta_3^{j_1} - \delta_2^{j_1}) &= (\delta_3^{j_2} - \delta_2^{j_2}) + k_3(p)\\ \Leftrightarrow\ & (\delta_3^{j_1} - \delta_2^{j_1}) &= (\delta_3^{j_2} - \delta_2^{j_2}) - k_2(p)\\ \Leftrightarrow\ & (\delta_2^{j_1} - \delta_1^{j_1}) &= (\delta_2^{j_2} - \delta_1^{j_2}) + k_2(p)\ .\end{aligned}$$

Consequently, the third transmission does not impose a new constraint on the adversary's location p_A if $\beta = 180°$.

Regarding the direction difference $\alpha_{j_1} - \alpha_{j_2}$, we can summarize that the closer the difference is to 180°, the higher the percentage of locations which could be securely verified after the third transmission. In fact, we also did simulations for $\alpha_{j_1} - \alpha_{j_2} > 180°$, but the results were identical to those for $360° - (\alpha_{j_1} - \alpha_{j_2})$.

We conclude from our simulations that with $\beta = 110°$ or $\beta = 250°$ and a direction difference of $|\alpha_{j_1} - \alpha_{j_2}| = 180°$, more than 93% of all location verification scenarios could be securely verified with two verifiers and three transmissions of the location claim. This is an improvement of 300% compared to mobility of opportunity.

3.3.4 Lessons Learned

The key insights from our simulations are the following heuristics:

- The speed of the verifiers plays only a minor role for the security of our scheme. However, this might change in real-world implementations when measurement noise is added to the system. In this case, larger steps provide larger MDTOA values and thus lower noise ratios.

- Never use β close to 0° or 180° since they provide poor to no security. Also avoid that verifiers move into the same direction ($|\alpha_{j_1} - \alpha_{j_2}| \approx 0$).
- To maximize the verification accuracy and speed, the best choices for β (and every further turn angle) are either around 110° or around 250°. If possible, the verifiers should move such that their initial direction difference $|\alpha_{j_1} - \alpha_{j_2}|$ is as close to 180° as possible. Given these conditions, two verifiers can securely verify over 93% of the locations in the area of interest with three transmissions of the location claim by the prover.
- To achieve a 100% detection rate under random attack scenarios, at least three verifiers ($|\mathcal{V}| \geq 3$) or four transmissions ($n \geq 4$) are required.

It is worth noting that all heuristics can be executed offline without active coordination between the verifiers. For example, to ensure that $|\alpha_{j_1} - \alpha_{j_2}| \approx 180°$, verifiers could choose opposite directions based on p and unique IDs without further communication.

3.4 Discussion

A key insight of our security analysis is that security can be significantly improved by controlling the verifiers' movements. More specifically, directed mobility provides *provable* security while mobility of opportunity requires more transmissions or verifiers to achieve *statistically* good security. On the other hand, mobility of opportunity provides better scalability and flexibility. Using specific movement patterns is a trade-off between both extremes which provides good security with few verifiers and low communication overhead. In practice, however, system designs and existing infrastructure usually dictate the mode of mobility that can be applied. For instance, while directed mobility provides the best security and is therefore preferable, it might not always be feasible due to its limited scalability or physical constraints.

We summarize that our protocol constitutes a significant step into the direction of *security through mobility*. Our results clearly demonstrate that security in the context of location awareness benefits from the mobility in terms of lower system requirements and communication overhead. Compared to other location verification schemes (see Table 2.2 on page 21), our protocol provides strong security and high accuracy while, at the same time, the minimum number of verifiers is low, no synchronization is necessary, and the protocol is completely passive. *To the best of our knowledge, there is no other location verification scheme achieving the same level of security with such an efficiency.* On the downside, however, the need for mobility could be interpreted as an expensive system requirement itself. In real-world scenarios, installing fixed sensors is often much easier and cheaper than using mobile nodes such as drones or aircraft. Nevertheless, with the ongoing proliferation of location aware mobile nodes, e.g., in transportation systems or drone delivery services, the level of mobility is constantly increasing and ignoring its benefits for security would be a waste of resources.

3.4.1 Stronger Adversaries

So far, we only considered a single adversary with one antenna. The technological progress, however, does not only provide advantages for the defending side. Adversaries also benefit from new sophisticated and accurate communication technologies. For instance, in [45], Moser et al. have recently demonstrated that coordinated location spoofing attacks on Time of Arrival (TOA)-based systems are already possible with commercial off-the-shelf software-defined radios. To be able to defend against such a strong multi-device attacker, our protocol has to be extended by more sophisticated means of spoofing detection. The authors of [45] propose using phase- and frequency-based features of the incoming signals to detect whether they come from the same source or from several different devices. In principle, this approach could be implemented alongside our protocol. It would, however, require the exchange of signal data between the verifiers and therefore reduce its “lightweightness”.

Whether such an extension of our protocol is really necessary in practice remains questionable and is subject to further research. We argue that mobility of verifiers provides security even beyond the single adversary model. A moving verifier is a much harder target for timing attacks such as those described in [45]. The adversary would have to track the passive verifiers’ movements at a very high precision to calculate the required timings accurately. Furthermore, depending on the distance between the verifiers and the location of the adversary, extremely accurate directed antennas would be required to transmit a signal only to the intended verifier. The probability that the adversary’s signal is only received by the intended verifier significantly decreases with an increasing density of moving verifiers.

The second limiting assumption made during our analysis is the adversary’s lack of mobility. A mobile adversary could change its location during the verification process such that its change in distances to the verifiers compensates for any inconsistencies left in the reception times after adjusting the intertransmission time. However, the adversary would have to change its location according to the behavior of the verifiers. As a consequence, it is forced to move along a certain path at a certain velocity, both not under its control. Although this is possible in theory, the adversary would certainly face physical limitations and obstacles in practice. An open research question is here how the location change required by an adversary behaves as a function of p_A and the verifiers' movements and whether such attacks are realistic given physical constraints such as maximum possible speeds. In summary, while the focus of this thesis lies on different ways to exploit the fundamental effects of mobility to improve security, future research could investigate more complex attack scenarios with mobile and distributed attackers.

A new idea comes suddenly and in a rather intuitive way. But intuition is nothing but the outcome of earlier intellectual experience.

— Albert Einstein in a letter to Dr. H. L. Gordon

4

Verification of Mobile Provers

Contents

4.1 Network & Threat Model 54
4.2 Verifying Tracks in the Time Domain 58
4.2.1 Security Analysis 59
4.2.2 Dealing With Noise 65
4.2.3 Evaluation 69
4.2.4 Realistic Movement Patterns 75
4.2.5 Discussion 81
4.3 Verifying Motion in the Frequency Domain 84
4.3.1 Security Analysis 85
4.3.2 Evaluation 88
4.3.3 Adapted Verification Scheme 92
4.3.4 Real-world Performance 98
4.3.5 Discussion 100
4.4 Outlook: Verifying Mobility in the Space Domain 101
4.4.1 Sanity Check Protocol 102
4.4.2 Lossy Communication Channel 104
4.4.3 Security Considerations 105

As we have demonstrated in the previous chapter, mobility improves the security of location awareness in systems with *stationary provers*. However, we have also seen in section 2.2 that no adequate methods exist to verify the spatial state of *mobile provers*. In fact, methods which rely on the exchange of several messages may not be applicable to mobile provers at all since the location to be verified would change between each transmission. Moreover, location verification schemes are per definition not suitable for moving provers since, depending on the prover's speed, verified locations may be obsolete just a moment later. As a consequence, new schemes are required that specifically take mobile provers into account. The problem statement needs to be adapted since the spatial state of mobile provers does not only consist of a single location. It also consists of the node's speed, moving direction, and a set of locations it has previously visited (track).

This chapter fills this gap. We first extend the problem of location verification towards mobility and then provide adequate methods to verify a mobile node's track and spatial state. We show that the effects of mobility in the time domain (MDTOA) and in the frequency domain (FDOA) provide sufficient means for designing solutions that are both secure and efficient.

4.1 Network & Threat Model

Verification schemes which consider mobile provers are particularly important for systems with high velocities. One reason for this is that, depending on the application, location information expires very quickly due to the rapid movements of nodes. A nicely illustrative example is Air Traffic Control (ATC) where locations are only valid for a few seconds since en route aircraft cover about 240 meters every second. In addition, air traffic controllers need both reliable instantaneous velocity information as well as historical track information to recognize whether aircraft are turning and to be able to extrapolate tracks to maintain the separation minima. Since it is a fitting and important example, ATC will serve as a motivation for this chapter and the following system model is specifically inspired by the Automatic Dependent Surveillance-Broadcast (ADS-B). For more details on ADS-B, see chapter 5. However, we emphasize that there are also other well conceivable areas of application for our scheme. The key characteristic of the following system model is the mobility of the prover. Therefore, any location aware application with mobile nodes (see subsection 2.1.2 for examples) might be a potential target system for our scheme.

In this chapter, we consider a moving prover P which periodically broadcasts its spatial state at transmission time t_i, i.e., $m_i = (\hat{t}_i, p_P(t_i), v_P(t_i))$ for $i = 1, \ldots, n$, to a set of stationary verifiers $\mathcal{V}$ using a wireless communication channel. Since we will only consider one prover for now, we omit the subscript P and denote the claimed location at a point in time t by $p(t)$ and the claimed velocity by $v(t)$. We assume that there is no compromised verifier and all verifiers are able to communicate securely with each other. We further assume that each verifier $V_j \in \mathcal{V}$ knows its exact position p_j. Note that we simply write p_j since the verifiers are stationary, i.e, $p_j(t) = p_j$ at all times t.

In contrast to the previous chapter, P's spatial state and its claims now include additional velocity information. While the literature provides a plethora of means to verify locations (section 2.2), velocity and visited locations ("track") have not received any attention so far. However, as the ATC example above demonstrates, reliable velocity and track information can be as important for some applications as locations are for others. For collision avoidance, whether automatic or manually through controllers, pure location information is not sufficient to predict a node's location in the future and identify a potential collision course. Accurate velocity information is needed to accomplish this reliably. Furthermore, knowledge about a node's recent track, that is, its recent movement behavior, helps controllers to identify whether aircraft are turning or not.

Based on the effects of mobility on wireless communications (section 2.3), we identify two different approaches to accounting for prover mobility by including it into the problem statement. More specifically, we separate the verification of mobility into verifying tracks and verifying velocities. We do so for several reasons. First and as we will see shortly, tracks and velocities are measured in different domains. This results in a natural demand for a separation of the two problems. Second, tracks and velocities may not both be considered by all applications. Some applications only consider tracks while others only rely on velocity information. Therefore, the separation will result in two independent methods which can, but do not have to be combined and thus, provide a maximum level of flexibility. We now continue with detailing the two (sub-)problems and providing the respective adversary models.

Secure Track Verification

In the time domain, the MDTOA provides us with means to jointly consider multiple *location* claims. It captures the mobility of the prover in terms of measurable distance changes between transmissions. Moreover, it does so without the need for time synchronization or extra communication. On the downside, however, it does not allow us to directly measure the prover's velocity. The MDTOA only provides us with "snapshots" of the prover's locations without including any velocity information. Since the prover can change its velocity, the mere distance between two reported locations and the duration between two transmissions does not allow us to reliably infer any information about the prover's speed and direction of movement. This implies that verification of the prover's velocity is not possible with the MDTOA. Nevertheless, it allows us to verify a sequence of locations the prover has visited previously, its so called *track*. More formally, a track $\mathcal{T}$ of a prover P is a record of P's claimed locations and the reported transmission timestamps, i.e., $\mathcal{T} = \{m_P, \ldots, m_P\}$. In light of the verification context, we call $\mathcal{T}$ a *track claim*. The respective verification problem is called *secure track verification* and is, in combination with the above system model, defined as follows:

> A set of *unsynchronized* and *stationary* verifiers $\mathcal{V}$ wish to verify whether a *mobile* prover P moves along a claimed track $\mathcal{T}$.

Track verification, as defined here, does not directly consider any velocity information and an efficient implementation of track verification schemes would omit the velocity information from messages m_i. Since a track claim is then technically a sequence of location claims from a single moving prover, track verification can be considered the logical extension of location verification towards mobile provers. As explained above, however, it does not allow us to verify velocity information. Therefore, we continue with the second complementary problem statement which explicitly considers velocity information.

Secure Motion Verification

Velocity is well measurable in the frequency domain through the Doppler effect. As explained earlier in section 2.3, the received frequency differs from the transmission frequency according to the sender's speed relative to the receiver. Hence, measuring the received frequency provides us directly with information about the senders velocity. Unfortunately, as we will show in later in this chapter, aircraft transponders have a rather poor transmission frequency accuracy and stability. This prevents us from measuring the Doppler effect with a single receiver since the exact transmission frequency is unknown. Yet, as also explained earlier in section 2.3, considering the Frequency Difference of Arrival (FDOA) instead of the absolute received frequency allows us to measure differences in radial speeds observed by several receivers without knowing the transmission frequency. We conclude that with the possibility to directly measure physical effects of velocities of mobile provers and given the above basic system model, we can define the second problem statement of *secure motion verification*as follows:

> A set of *stationary* verifiers $\mathcal{V}$ wish to verify whether a *moving* prover P's reported spatial state m_i is accurate.

Note that we preferred the term “motion verification” over “velocity verification” to emphasize that the prover has to be in motion to be eligible for this problem. In contrast, stationary provers also have a velocity, although it is zero, but do not fit to the problem considered here. A further noteworthy observation is that *motion claims* m_i are combinations of location claims, claimed directions of movement, and claimed speeds. In that sense, the problem of location verification can be considered a subproblem of secure motion verification, that is, any solution to the problem of motion verification also solves location verification. This also implies that the track verification problem can be solved by a “point-wise” motion verification scheme. If each location claim on a track is verified using motion verification, the track itself is secure. However, the strength of track verification is its use of the MDTOA and the resulting simplicity, lightweightness, and flexibility. Since TOA is a common measure for many localization and other verification techniques, most available hardware provides the accurate timestamps needed to determine the MDTOA. On the contrary, measuring the frequency of arrival is much less common and is not supported by most available receivers out of the box.

Threat Model

The natural enemy to both track verification and motion verification is a malicious node claiming false locations and/or velocities. In particular, we consider a single stationary adversary A located at position p_A. We assume that it uses an omni-directional antenna to broadcast the motion or track claim. This assumption ensures that all verifiers receive the exact same location claims during the verification process. As in chapter 3, we assume that the adversary also knows the exact positions of all verifiers. Further adversarial models such as mobile adversaries or adversaries with limited knowledge are discussed at the end of the next section.

Since we assume a stationary adversary claiming to be in motion, A's violation of the truth is inherently provided. For completeness, however, we note that in the case of track verification, A tries to claim a false track, i.e., $p_A \neq p_i$ for at least one $m_i \in \mathcal{T}$. In case of motion verification, the adversary either tries to spoof a location $p(t) \neq p_A(t)$ or a velocity $v(t) \neq v_A(t)$ or both at the same time.

With respect to the ATC scenario, a real-world encounter of our threat model could be an adversary positioned on ground next to an airport while injecting fake position or velocity reports to cause confusion or prevent departures. As we will demonstrate in chapter 5, such attacks are feasible even with low-cost hardware.

4.2 Verifying Tracks in the Time Domain

Based on the definition of the MDTOA (Equation 2.1 on page 23), we can conclude that for accurate track claims from an honest prover P, the interarrival time of two broadcasts m_{i-1} and m_i at verifier $V_j \in \mathcal{V}$ differs from the intertransmission time by the difference in propagation delays between P's locations at transmission times and V_j's location:

$$t_{i,j} - t_{i-1,j} = t_i - t_{i-1} + \left(\frac{\|p_j - p(t_i)\| - \|p_j - p(t_{i-1})\|}{c} \right) ,$$

or, expressed in node-local timestamps,

$$\hat{t}_{i,j} - \hat{t}_{i-1,j} = \hat{t}_i - \hat{t}_{i-1} + \left(\frac{\|p_j - p(t_i)\| - \|p_j - p(t_{i-1})\|}{c} \right) .$$

With the notation from the previous chapter, i.e.,

- $\hat{\Delta}^j_{i-1,i} = \hat{t}_{i,j} - \hat{t}_{i-1,j}$ denotes the interarrival time measured by V_j,
- $\hat{\Delta}^j_{i-1,i} = \hat{t}_i - \hat{t}_{i-1}$ denotes the intertransmission time claimed by P, and
- $\hat{\delta}^j_i = \frac{\|p_j - p(t_i)\|}{c}$ denotes the estimated propagation delay of the i-th transmission between P and V_j,

we can formulate the same verification condition as before:

$$\hat{\Delta}^j_{i_1,i_2} \quad \overset{?}{=} \quad \hat{\Delta}_{i_1,i_2} + \left(\hat{\delta}^j_{i_2} - \hat{\delta}^j_{i_1} \right) . \tag{4.1}$$

Our basic track verification procedure works as follows. A prover P periodically broadcasts its location claims m_i ($i = 1, 2, \dots$) while moving along a track. Each verifier $V_j \in \mathcal{V}$ learns the track by receiving and recording all m_i along with the reception timestamps $\hat{t}_{i,j}$. For each received m_i with $i > 1$, verifier V_j also checks the *local* verification condition provided in Equation 4.1. If the condition is satisfied, the verifier remains silent. If, however, the expected interarrival time is not equal to the measured one, the verifier raises an alarm. A track is successfully verified if a least number of location claims n have been broadcasted and no alarm has been raised.

While this verification procedure is almost equal to that of the location verification scheme in chapter 3, there is a significant difference which has a positive effect on the security. In Equation 4.1, the time-dependent position is that of the prover, not of the verifier as in the location verification scheme (Equation 3.1). In principle, we reformulated the location verification problem by moving the mobility to the prover's side of the equation, while the rest just stays the same. As we will see in the security analysis, this enables us to better analyze the problem formally since each additional verifier only introduces one new position, independent from the number of transmissions. In contrast, in the previous chapter, all verifiers contributed to the set of equations with one additional location per transmission, whereas here, only one node (the prover) brings in a new position with each transmission.

The protocol overhead and completeness of our track verification scheme are the same as those of the location verification scheme in chapter 3. For completeness, we repeat that the computational overhead for each verifier is in $\mathcal{O}(n)$, extra communication by the verifiers is not necessary as long as there is no alarm, and the protocol is complete by the definition of the MDTOA. For more details about these properties, refer to subsection 3.2.2. Concerning the security property of our verification scheme, we claim that given a certain number of verifiers or transmissions, a dishonest prover cannot send false location claims without violating Equation 4.1 for at least one verifier. To prove this hypothesis, we conduct a theoretical security analysis next.

4.2.1 Security Analysis

For our analysis, we assume that the adversary's goal is to claim a track with two location claims $m_1 = (\hat{t}_1, p(t_1))$ and $m_2 = (\hat{t}_2, p(t_2))$ with $p(t_1) \neq p(t_2)$. We can do so without loss of generality, since Equation 4.1 constitutes a pairwise check for all claims in $\mathcal{T}$ without particular order. Hence, if our scheme is secure for arbitrary m_1 and m_2, it is also secure for track $\mathcal{T}$. To provide a better understanding how security is established, we analyze our scheme step by step by increasing the number of verifiers $|\mathcal{V}|$. However, since $|\mathcal{V}| = 1$ is equal to the first case of the location verification scheme (see section 3.3), we skip the details of this case here and just recapitulate that more than one verifier is required since the adversary can otherwise spoof arbitrary tracks by simply adjusting the transmission times accordingly.

Case $|\mathcal{V}| = 2$

The adversary tries to forge a location claim by adapting the real transmission times t_i such that its location claims seem honest for the two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ located at p_{j_1} and p_{j_2}. Having two verifiers, the adversary's signal experiences two independent propagation delays $\delta_i^{j_1}$ and $\delta_i^{j_2}$ and both verifiers expect independent propagation delays $\hat{\delta}_i^{j_1}$ and $\hat{\delta}_i^{j_2}$. As a result, the adversary has to find an intertransmission interval $\Delta_{i-1,i}$ such that the local verification check (Equation 4.1) is satisfied for both verifiers. Hence, the following system of equations must be satisfied:

$$\begin{aligned}\hat{\Delta}^{j_1}_{i-1,i} &= \hat{\Delta}_{i-1,i} + \left(\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}\right)\\ \hat{\Delta}^{j_2}_{i-1,i} &= \hat{\Delta}_{i-1,i} + \left(\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}\right) .\end{aligned}$$

Since verifiers and adversary are all stationary, the interarrival time can be modeled by

$$\hat{\Delta}^{j}_{i-1,i} = \Delta_{i-1,i} + \underbrace{(\delta_i^{j} - \delta_{i-1}^{j})}_{=0} = \Delta_{i-1,i} .$$

Plugging this into the above system of equations and isolating $\Delta_{i-1,i}$ yields

$$\begin{aligned}\Delta_{i-1,i} &= \hat{\Delta}_{i-1,i} + \left(\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}\right)\\ \Delta_{i-1,i} &= \hat{\Delta}_{i-1,i} + \left(\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}\right) .\end{aligned}$$

As a result, the adversary is limited to claiming $p(t_1)$ and $p(t_2)$ which satisfy

$$\left(\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}\right) = \left(\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}\right) ,$$

or, in terms of distances and rearranged with respect to the transmission times,

$$\|p_{j_1} - p(t_{i-1})\| - \|p_{j_2} - p(t_{i-1})\| = \|p_{j_1} - p(t_i)\| - \|p_{j_2} - p(t_i)\| . \tag{4.2}$$

Let us now assume that the adversary has chosen an arbitrary $p(t_i)$. According to the previous results, the next claimed position $p(t_{i+1})$ must be in the following set of points:

$$\mathcal{H}(p(t_i), p_{j_1}, p_{j_2}) = \left\{ \overline{p} \in \mathbb{R}^d \;\middle|\; \begin{aligned} &\|p_{j_1} - p(t_i)\| - \|p_{j_2} - p(t_i)\| = \\ &\|p_{j_1} - \overline{p}\| - \|p_{j_2} - \overline{p}\| \end{aligned} \right\} .$$

where d is the number of dimensions. In words, the distance difference between $p(t_{i+1})$ and the two verifiers' locations must be equal to that of $p(t_i)$ to the verifiers. In the two-dimensional case, this set of positions $\mathcal{H}(p(t_i), p_{j_1}, p_{j_2})$ corresponds to one arm of a hyperbola with foci p_{j_1} and p_{j_2} and a difference of distances to the foci of

$$\|p_{j_1} - p(t_i)\| - \|p_{j_2} - p(t_i)\| \ .$$

With $d = 3$, $\mathcal{H}$ is one sheet of a hyperboloid with the same parameters.

The key insight of this result is that the adversary cannot claim arbitrary tracks anymore. In particular, it loses one degree of freedom with the introduction of a second verifier. It is limited in its choice for the $i+1$-th location claim to $p(t_{i+1}) \in \mathcal{H}(p(t_i), p_{j_1}, p_{j_2})$. However, the adversary can still spoof tracks that go through an arbitrary position of interest. Although this might be sufficient for some attacks, being restricted to a hyperbola before and after claiming the position of interest is already a significant limitation. Furthermore, the two verifiers can easily check whether the locations of a claimed track lie on such a hyperbola. In case they do, they can consider the track being suspicious. In scenarios where hyperbolic tracks are impossible (e.g., roads in a vehicular network), attacks would not remain undetected and *two verifiers are sufficient to securely verify tracks.*

Case $|\mathcal{V}| = 3$

We can extend the constraint of the previous case to

$$\left(\hat{\delta}_i^{j_1} - \hat{\delta}_{i-1}^{j_1}\right) = \left(\hat{\delta}_i^{j_2} - \hat{\delta}_{i-1}^{j_2}\right) = \left(\hat{\delta}_i^{j_3} - \hat{\delta}_{i-1}^{j_3}\right)$$

for two location claims $m_{i-1}(t_{i-1})$ and m_i and three verifiers $V_{j_1}, V_{j_2}, V_{j_3} \in \mathcal{V}$. Analogously to the previous case, this constraint can only be satisfied by an adversary if it forges the location claims such that the pairwise hyperbolas (or hyperboloids in 3D) of the three verifiers all intersect at $p(t_{i-1})$ and $p(t_i)$. More formally, after $p(t_{i-1})$ has been claimed, $p(t_i)$ must satisfy the following constraint:

$$p_i(t_i) \in \mathcal{H}(p(t_{i-1}), p_{j_1}, p_{j_2}) \ \cap \ \mathcal{H}(p(t_{i-1}), p_{j_1}, p_{j_3}) \ \cap \ \mathcal{H}(p(t_{i-1}), p_{j_2}, p_{j_3}) \ .$$

It is obvious that, by continuing this method and adding more verifiers, we can generalize this constraint to arbitrary sets of verifiers $\mathcal{V}$ with $|\mathcal{V}| > 1$ and arbitrary $i > 1$:

$$p(t_i) \in \bigcap_{\{V_{j_1}, V_{j_2}\} \in \mathcal{V}} \mathcal{H}(p(t_{i-1}), p_{j_1}, p_{j_2}) \ . \tag{4.3}$$

From the adversary's point of view this result means that once it has claimed the first position $p(t_1)$, all subsequent position claims of track $\mathcal{T}$ must lie on all pair-wise hyperbola arms (or hyperboloid sheets), or, in other words, on their intersections. That means that by adding a third verifier to the system, the adversary's degree of freedom is reduced from hyperbolas to intersections of hyperbolas.

We now analyze these intersections. For the sake of concise presentation, we only consider the two-dimensional case. Extending our results to three dimensions is straightforward by considering intersections of hyperboloids instead of hyperbolas.

Let $d(i, j_1, j_2)$ be the difference between the distances from the i-th claimed location to V_{j_1} and V_{j_2}, i.e.,

$$d(i, j_1, j_2) = \|p_{j_1} - p(t_i)\| - \|p_{j_2} - p(t_i)\| \ .$$

We can plug this into Equation 4.2 and obtain the following system of equations where solutions $p = (x, y) \in \mathbb{R}^2$ are by definition the intersections of $\mathcal{H}(p(t_{i-1}), p_{j_1}, p_{j_2})$ and $\mathcal{H}(p(t_{i-1}), p_{j_1}, p_{j_3})$:

$$\begin{aligned} \|p_{j_1} - p\| - \|p_{j_2} - p\| &= d(i, j_1, j_2) \\ \|p_{j_1} - p\| - \|p_{j_3} - p\| &= d(i, j_1, j_3) \ . \end{aligned}$$

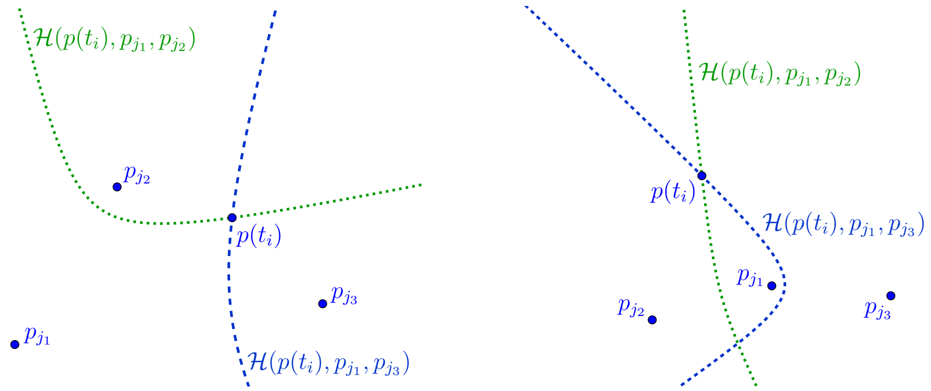
By using the notation (x_j, y_j) for positions $p_j \in \mathbb{R}^2$ and by applying the definition of the Euclidean norm, this system can be rewritten as:

$$\begin{aligned} \sqrt{(x_{j_1} - x)^2 + (y_{j_1} - y)^2} - \sqrt{(x_{j_2} - x)^2 + (y_{j_2} - y)^2} &= d(i, j_1, j_2) \\ \sqrt{(x_{j_1} - x)^2 + (y_{j_1} - y)^2} - \sqrt{(x_{j_3} - x)^2 + (y_{j_3} - y)^2} &= d(i, j_1, j_3) \ . \end{aligned}$$

Squaring and rearranging these equations yields

$$\sqrt{(x_{j_1} - x)^2 + (y_{j_1} - y)^2} = x \cdot c_1 + y \cdot c_2 + c_3 \tag{4.4}$$

$$\sqrt{(x_{j_1} - x)^2 + (y_{j_1} - y)^2} = x \cdot c_4 + y \cdot c_5 + c_6 \tag{4.5}$$


$\mathcal{H}(p(t_i), p_{j_1}, p_{j_2})$
p_{j_2}
$p(t_i)$
p_{j_3}
p_{j_1}
$\mathcal{H}(p(t_i), p_{j_1}, p_{j_3})$
$\mathcal{H}(p(t_i), p_{j_1}, p_{j_2})$
$p(t_i)$
$\mathcal{H}(p(t_i), p_{j_1}, p_{j_3})$
p_{j_1}
p_{j_2}
p_{j_3}

(a) Example with one intersection. The adversary cannot claim any further locations without being detected.

(b) Example with two intersections. The adversary could claim the second intersection without being detected.

Figure 4.1: Two example scenarios with three verifiers, a claimed position $p(t_i)$, and two resulting hyperbolas.

with constants

$$\begin{aligned}
c_1 &= (x_{j_2} - x_{j_1})/d(i, j_1, j_2) \\
c_2 &= (y_{j_2} - y_{j_1})/d(i, j_1, j_2) \\
c_3 &= (x_{j_1}^2 + y_{j_1}^2 - x_{j_2}^2 - y_{j_2}^2 - d(i, j_1, j_2)^2)/(2 \cdot d(i, j_1, j_2)) \\
c_4 &= (x_{j_3} - x_{j_1})/d(i, j_1, j_3) \\
c_5 &= (y_{j_3} - y_{j_1})/d(i, j_1, j_3) \\
c_6 &= (x_{j_1}^2 + y_{j_1}^2 - x_{j_3}^2 - y_{j_3}^2 - d(i, j_1, j_3)^2)/(2 \cdot d(i, j_1, j_3)) \; .
\end{aligned}$$

Subtracting Equation 4.5 from Equation Equation 4.4 results in

$$y = x \cdot \frac{c_1 - c_4}{c_5 - c_2} + \frac{c_3 - c_6}{c_5 - c_2} \; .$$

Plugging this equation into one of the initial equations results in a quadratic equation for x and y. Quadratic equations have either zero, one, or two solutions. In our case, we know already that by construction of the hyperbolas, the system has at least one solution $p(t_{i-1})$. Thus, there is either no or at most one additional position left for the adversary to spoof on a track without violating Equation 3.1 for one of the verifiers.

Two example scenarios with three verifiers and their pairwise hyperbolas are depicted in Figure 4.1. The adversary wants to spoof a track and claims to be at $p(t_i)$. In the left scenario (Figure 4.1a), there is no further intersection of the hyperbolas and thus, the adversary cannot claim a second position without being detected by at least one verifier. In Figure 4.1b, the adversary can claim exactly one more location (the second intersection) without being detected.

It is worth noting that adding the hyperbola $\mathcal{H}(p(t_{i-1}), p_{j_2}, p_{j_3})$ to the two scenarios shown in Figure 4.1 will not change the situations. The reason for this is that this third hyperbola is redundant and does not add any further restrictions since it is defined by the same distances as the other two hyperbolas.

Case $|V| > 3$

Equation 4.3 is a general result which also holds for more than three verifiers. For the two-dimensional case, the guarantees given by three verifiers are already sufficient since attacks using tracks with two intersections can simply be prevented by requiring $|\mathcal{T}| \geq 3$. However, more than three verifiers can be beneficial to mitigate noise in the verification data such as measurement errors, clock drifts, or position errors. This interesting issue of imperfect verification data and how to use $|\mathcal{V}| > 3$ to improve the accuracy of track verification is subject of the next sections.

On a final note, in three dimensions, our problem is very similar to hyperbolic localization methods based on time-difference of arrival measurements (e.g. multilateration). A fourth verifier would be necessary to pin the adversary down to a single position. In general, $|\mathcal{V}|$ verifiers result in $|\mathcal{V}| - 1$ independent hyperboloids. With three-dimensional locations and $|\mathcal{V}| = 3$, the intersections of the two resulting hyperboloids form a curve. As in the two-dimensional case, adding a fourth verifier reduces the number of intersections to at most two points in space.

Conclusions from the Analysis

The above analysis shows that the adversary loses one degree of freedom with each additional verifier. The intuition behind this is as follows. As the honest prover is changing its position between individual location claims, the propagation delays to each verifier must also change in order to satisfy Equation 4.1 for all verifiers. Thus, adversaries would have to vary the propagation delays to each of the verifiers independently to successfully pretend movement. Since all verifiers receive the same messages (due to the broadcast transmission), this is not possible. As a result, the only spoofable track for a stationary adversary is the track on which the differences in propagation delays to each verifier are constant. For two verifiers, this is a hyperbola. For more than two verifiers, this property only holds for the intersections of the pairwise hyperbolas (see Figure 4.1).

To conclude the analysis, we generalize the assumptions for secure track verification as follows. Let d be the number of dimensions, that is $p(t_i) \in \mathbb{R}^d$. Then our scheme detects track spoofing attacks if any of the following two conditions is met:

- There are two locations claims $m_{i_1}, m_{i_2} \in \mathcal{T}$ and two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ such that $p(t_{i_2}) \notin \mathcal{H}(p(t_{i_1}), p_{j_1}, p_{j_2})$ holds.
- The number of location claims on the track is $|\mathcal{T}| \geq 3$ and the number of verifiers is $|\mathcal{V}| \geq d+1$.

4.2.2 Dealing With Noise

In practice, verifiers have to deal with imperfect verification data since time and position measurements are prone to errors. For instance, clocks have different speeds which results in non-negligible drifts over time. In order to assess the practicality and performance of our verification scheme under realistic conditions, we extend our track verification scheme with an error model and evaluate its feasibility under realistic conditions.

Error Model

Clock Drift: The speed of clocks is highly dependent on environmental conditions such as pressure or temperature [46]. However, we assume that the duration of the verification process is in the order of seconds or minutes. Most environments (such as the interior of vehicles) are sufficiently stable within such time periods. We therefore assume that clock drift is linear and thus increases at a constant rate during the verification process. In accordance to that, we model clock drift as follows. The error due to clock drift linearly depends on the duration between two time measurements. It can be modeled by a drift coefficient t^j_{drift} for verifier $V_j \in \mathcal{V}$. Assuming that V_j wants to measure an interarrival time $\Delta^j_{i-1,i}$, the clock drift error $\epsilon^j_{\mathit{drift}}$ of V_j's measurement $\hat{\Delta}^j_{i-1,i} = \hat{t}_i - \hat{t}_{i-1}$ is modeled by

$$\epsilon^j_{\mathit{drift}} = \Delta^j_{i-1,i} \cdot t^j_{\mathit{drift}} \ .$$

Note that for a perfect clock, $t^j_{\mathit{drift}} = 1$ holds.

Measurement & Channel Noise: Measuring points in time at which events occur always involves measurement errors. For instance, systems are clocked by oscillators running at certain rates. Components only perform actions if a pulse or pulse edge of the oscillator is present. As a consequence, observations can only be made at *discrete* points in time. This leads to measurement errors when events of interest (such as the arrival of a signal) occur between two clock ticks. Besides timing errors, wireless transmission characteristics such as multipath propagation distort the signal. This may also results in noise when determining timestamps for signal arrivals. Moreover, our scheme may also suffer from erroneous position information. If provers use GPS to determine their positions, the location claims may contain errors of up to 15 m.

We assume that measurement and channel noise are independent for each location claim and each of its associated timestamps. In accordance with [41], we summarize all sources of noise in a zero-mean Gaussian random variable $\epsilon \sim \mathcal{N}(0, \sigma^2)$. The variance σ^2 depends on the accuracy of the system components involved in the verification process. For example, if clocks with higher rates are used, timestamps become more accurate and σ^2 becomes smaller.

By combining clock drift and noise, we conclude that the measured interarrival time can be modeled by

$$\hat{\Delta}^j_{i-1,i} = \Delta^j_{i-1,i} + \epsilon^j_{drift} + \epsilon = \Delta^j_{i-1,i} \cdot (1 + t^j_{drift}) + \epsilon \ . \tag{4.6}$$

In order to account for this noise during the verification, we extend our scheme in two different ways and propose a local and global variant of our track verification protocol. In local verification, verifiers calculate and check their verification results locally. This has the advantage that they do not have to communicate with each other. As a result, communication overhead is minimal and verifiers do not have to be connected. The only communication required by the verifiers is sending an alarm to a central entity in case of an attack. This simplicity, however, comes at a price. Local verification does not take full advantage of the total number of verifiers when it comes to noise cancellation. Therefore, we also propose a global scheme, which is based on the local scheme but verifiers collaborate in order to reduce the impact of noise. This allows engineers to trade higher resilience to noise for communication overhead and vice versa.

Local Track Verification Scheme

The noise in real systems renders a direct check of Equation 4.1 to verify a track infeasible. Therefore, we adapt our basic verification scheme to deal with noisy values. The idea is to use all received location claims to estimate the error. As shown in [41], jointly estimating clock drift and measurement error is not feasible since the Cramer-Rao lower bound of the estimation error is too high. Therefore, we perform our verification in two steps.

The first step estimates the clock drift. This estimate is then used in the second step to subtract ϵ^j_{drift} from our measurements. Given a track claim $\mathcal{T}$, the clock drift coefficient t^j_{drift} of V_j can then be estimated with

$$\hat{t}^j_{drift} = \frac{1}{\binom{|\mathcal{T}|}{2}} \sum_{i_1=1}^{|\mathcal{T}|-1} \sum_{i_2=i_1+1}^{|\mathcal{T}|} \left(\frac{\hat{\Delta}^j_{i_1,i_2}}{\hat{\Delta}_{i_1,i_2} + (\hat{\delta}^j_{i_2} - \hat{\delta}^j_{i_1})} - 1 \right) \ . \tag{4.7}$$

From a security perspective, estimating the clock drift in this way raises the question whether an adversary can take advantage of pretending certain clock drifts or not. The answer to that question is no. Since we do not make any assumptions on clock drifts, a fake clock drift is just as good as a true one, and both will be equally eliminated by Equation Equation 4.7. Faking different clock drifts during one track is even worse, since the estimation error will be high and, thus, increase the difference between expected and measured interarrival times (which leads to a rejection of the claimed track). Hence, fake clock drifts do not pose a threat to our scheme.

The verification is finally done in a second step by calculating the mean squared error when subtracting the left-hand side from the corrected right-hand side of Equation 4.1. We denote the *local verification result* of verifier V_j for a track $\mathcal{T}$ by $\varphi_j(\mathcal{T})$ and it is defined as

$$\varphi_j(\mathcal{T}) = \frac{1}{\binom{|\mathcal{T}|}{2}} \sum_{i_1=1}^{|\mathcal{T}|-1} \sum_{i_2=i_1+1}^{|\mathcal{T}|} \left((\hat{\Delta}_{i_1,i_2} + (\hat{\delta}^j_{i_2} - \hat{\delta}^j_{i_1})) \cdot (1 + \hat{t}^j_{drift}) - \hat{\Delta}^j_{i_1,i_2} \right)^2 \ . \tag{4.8}$$

The results of our security analysis in subsection 4.2.1 imply that for honest track claims, $\varphi_j(\mathcal{T})$ should converge to the average squared measurement error. For dishonest claims, $\varphi_j(\mathcal{T})$ must be higher for at least one verifier due to deviations caused by its dishonesty. In our track verification scheme, each verifier $V_j \in \mathcal{V}$ calculates $\varphi_j(\mathcal{T})$ and checks whether it is below a predefined threshold. In case a verifier's local result is higher than the threshold, the verification fails and the track is considered to be dishonest. We call this verification process *local track verification* as each verifier calculates its verification result locally without interacting with other verifiers.

The threshold for the local verification result is denoted by θ_{local}. It should be chosen based on the variance σ^2 of the measurement error ϵ and the number of location claims

$|\mathcal{T}|$ that are used for verification. As $|\mathcal{T}|$ increases, $\hat{t}^j_{drift}$ becomes more accurate and $\varphi_j(\mathcal{T})$ is supposed to converge to a value close to zero. An optimal θ_{local} must fulfill the same properties as a location verification scheme according to [29]:

- **Completeness:** If $\mathcal{T}$ is an honest track claim, $\varphi_j(\mathcal{T}) < \theta_{local}$ must hold for all verifiers $V_j \in \mathcal{V}$.
- **Security:** If T is a false track claim, $\varphi_j(\mathcal{T}) \geq \theta_{local}$ must hold for at least one verifier if the protocol is used in accordance with the results from subsection 4.2.1.

With an optimal threshold, the local verification scheme would be able to perfectly distinguish between honest and dishonest track claims. For later analyses and optimizations, we measure the "optimality" of θ_{local} and our system in terms of *false rejection* and *false acceptance* rates. A false rejection of a track means the detection of an attack, although the prover is honest. Conversely, a false acceptance occurs if a inaccurate track claim is not rejected by the system. Both rates can be controlled with θ_{local}. On the one hand, if θ_{local} is smaller than the highest possible $\varphi_j(\mathcal{T})$ for honest tracks, false rejections can occur. On the other hand, false acceptances are possible if θ_{local} is greater than the smallest possible $\varphi_j(\mathcal{T})$ for false track claims.

Global Track Verification Scheme

Bad hardware accuracy and verifier placements may result in false rejections and acceptances in our local verification scheme. The local verification scheme, however, does not take advantage of the total number of verifiers and their combined knowledge since it only considers local results. Higher numbers of verifiers can provide a more robust verification decision by combining all local results instead of considering them separately. We call this extension *global track verification.*

In our global verification scheme, the verifiers exchange their verification results $\varphi_j(\mathcal{T})$. It is worth noting that a even more robust approach could be designed by exchanging all measurement values since $\varphi_j(\mathcal{T})$ is a lossy abstraction. However, we decided to accept this loss of information to reduce communication overhead and keep our scheme lightweight. Based on the exchanged local verification results, each verifier (or a predetermined leader) calculates the global average verification result:

$$\varphi(\mathcal{T}) = \frac{1}{|\mathcal{V}|} \cdot \sum_{V_j \in \mathcal{V}} \varphi_j(\mathcal{T}) \ . \tag{4.9}$$

Similar to θ_{local}, we can define a threshold θ_{global} for $\varphi(\mathcal{T})$. A track $\mathcal{T}$ is accepted by the global verification scheme if $\varphi(\mathcal{T}) < \theta_{global}$. Accordingly, it is rejected if $\varphi(\mathcal{T}) \geq \theta_{global}$.

Global Vs. Local Verification

The choice whether to use the local or the global verification scheme depends on hardware constraints and infrastructure. In case it is cheaper to distribute many low-cost verifiers instead of a few high-end devices, the global verification is preferable. If verifiers are equipped with very accurate hardware, the local check might be the better choice as it is more sensitive to anomalies. Besides that, the local verification scheme produces less communication overhead and does not require a fully connected network of verifiers.

To combine the benefits of both approaches, a hybrid method could be used where verifiers only use the global method in cases where the local method cannot provide reliable decisions. More specifically, a hybrid method would choose a pessimistic θ_{local} and, in case of rejection, use the global verification scheme to support the decision. In this way, communication overhead could be reduced while maintaining the global scheme's accuracy.

4.2.3 Evaluation

In this section, we provide insights on the requirements, performance, and security of our noise-tolerant approach. We conducted extensive simulations and analyzed the effect of measurement error, clock drift, and number of claims on the verification result. To draw conclusions on the security (i.e., on false rejection and false acceptance rates), we compare the verification results of honest and dishonest track claims.

In order to keep the detection time low, it is necessary to keep the number of required messages ($|\mathcal{T}|$) as small as possible. Therefore we assume that the drift estimator $\hat{t}^j_{drift}$ is calculated with the same set of claims as the verification value $\varphi_j(\mathcal{T})$. As a result, they are not independent and since $\hat{t}^j_{drift}$ is used to calculate $\varphi_j(\mathcal{T})$, the error propagation in our scheme is complex and hard to analyze formally. Although we know that the variance of $\hat{t}^j_{drift}$ can be estimated with

$$Var(\hat{t}^j_{drift}) = \frac{\sigma^2}{\sum_{i_1=1}^{|\mathcal{T}|-1} \sum_{i_2=i_1+1}^{|\mathcal{T}|} \left(\hat{\Delta}_{i_1,i_2} + (\hat{\delta}^j_{i_2} - \hat{\delta}^j_{i_1})\right)^2 \cdot \binom{|\mathcal{T}|}{2}^2}$$

Parameter	**Description**
r	The radius of the circular area around the verifier
m	The number of messages per track, i.e. $m = \|\mathcal{T}\|$
v	The number of verifiers that receive the provers location claims, i.e. $v = \|\mathcal{V}\|$
σ	The standard deviation of the measurement error
σ_{drift}	The standard deviation of the random clock drift coefficients t^j_{drift} of verifiers V_j
Constant	**Description/Value**
c	The propagation speed of the signal is fixed to the speed of light (299792458 m/s)

Table 4.1: Overview on the simulation parameters for the error propagation analysis.

and the average estimation error converges to zero with increasing $|\mathcal{V}|$, we cannot set up a trivial error model for $\varphi_j(\mathcal{T})$ analogously. To analyze the error propagation nevertheless, we implemented the local and global verification schemes as a discrete-event simulation.

Simulation Setup

Initially, we assigned a random clock drift t^j_{drift} to each verifier $V_j \in \mathcal{V}$. We drew t^j_{drift} from a zero-mean Gaussian distribution with standard deviation σ_{drift}. The signal propagation speed is fixed to the speed of light (299792458 m/s) for all simulations. To cancel effects caused by tracks with special properties (e.g., errors due to a bad dilution of precision), the prover moves on random tracks for this analysis. The location claims for each track are randomly chosen from a circular area `R` with radius r around the verifier's position. The prover's maximal change in distance to the verifier (and thus the mobility-differentiated time of arrival which is considered by our verification scheme) is limited by r. The unit for distances is meters, points in time and time periods are in seconds. Our simulation parameters and constants are summarized in Table 4.1.

A key result of our formal security analysis in subsection 4.2.1 is that an implementation of our scheme must always ensure that the area of interest `R` is always covered by at least three verifiers. If this is the case, we know that for at least one verifier V_j, the measured period $\hat{\Delta}^j_{i-1,i}$ differs from the expected interarrival time. To produce valid insights on the security of our scheme, we are particularly interested in the local verification value $\varphi_j(\mathcal{T})$ of this verifier. For this purpose and without loss of generality, we consider only one verifier in each simulation run and assume that it is the one whose verification condition is violated. We generate the deviation of the adversary's signal arrival times

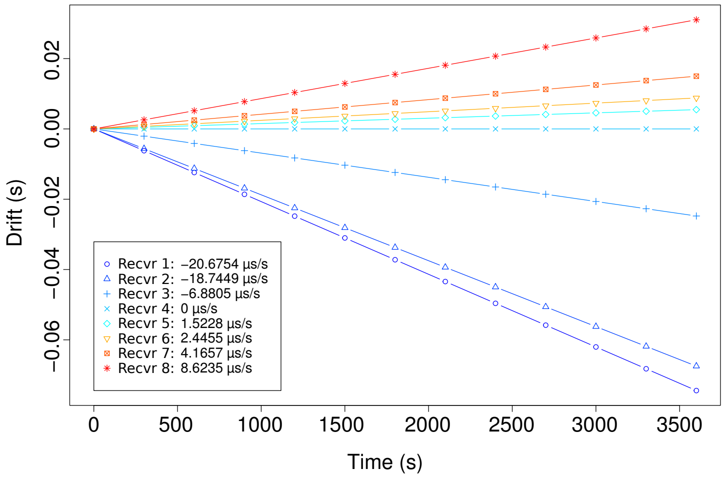


Figure 4.2: Estimated clock drifts of 8 SBS-3 receivers of the OpenSky Network over one hour. The drifts are shown relative to the clock of receiver 4.

from those of the honest prover by simply putting the adversary at a random but fixed position in R. The magnitude of the deviations can be controlled by r (larger r lead to larger deviations).

Clock Drift & Measurement Error

To keep our simulations realistic, we had to find appropriate parameters for our error model. With regard to our later analysis of the scheme's performance in a realistic air traffic scenario, we choose σ and σ_{drift} based on experiences from the OpenSky Network. The network uses low-cost receivers that are distributed to volunteers. The receivers collect (among other things) position reports that are periodically broadcasted by aircraft. They provide timestamps with a 50 ns resolution for the arrival of position reports. Besides that, most aircraft are using GPS to determine their positions. The typical position accuracy of GPS is about 15 m. In total, these noise sources lead to an estimation error of propagation delays of about 50 ns. Therefore, choosing $\sigma = 50$ ns for the measurement error seems appropriate.

It is worth mentioning that this is a rather pessimistic assumption. The OpenSky Network is using low-cost receivers which are not equipped with particularly good clocks. Better devices would produce timestamps with higher precisions. Furthermore, the US Federal Aviation Administration (FAA) is implementing navigation systems for civil aviation which can reduce positioning error to less than a meter [47].

To determine an appropriate standard deviation for clock drift errors, we considered the drifts of OpenSky's receivers relatively to each other. All considered receivers were

Kinetic Avionic's SBS-3 devices. To determine the clock drifts of the sensors, we used position reports received by multiple stations. By subtracting the difference in propagation delays to each receiver from the reception timestamps, we were able to obtain the offsets of the clocks over time and thus, the clock drift. We observed the clock drifts of eight receivers over a period of one hour and we found that they were constantly linear during that period. At this point, it is important to note that most receivers are indoors and not exposed to extreme temperature variations. The results of this analysis are shown in Figure 4.2. Accordingly, we choose a pessimistic standard deviation of $\sigma_{drift} = 20$ µs per second for the clock drifts of the verifiers.

Results: Local Verification

We first look at the local verification scheme as it is the basis for the global scheme. The goal of this analysis is twofold. On the one hand, we want to determine the least number of location claims needed to verify a track under the above error model. On the other hand, we are also interested in the benefits of receiving more location claims than actually needed. Ideally, the difference in $\varphi_j(\mathcal{T})$ between honest and dishonest tracks becomes more distinctive with each additional location claim as the estimators of our scheme become more accurate.

To draw conclusions from local simulation results about the overall performance of our verification scheme, we compare the maximum $\varphi_j(\mathcal{T})$ of 1000 honest tracks with the minimum $\varphi_j(\mathcal{T})$ of 1000 dishonest tracks. In other words, we check whether the worst verification result of honest tracks is greater than the best verification result of dishonest tracks. If this is the case, we can conclude that θ_{local} does not exist since we cannot perfectly distinguish honest from dishonest tracks. Let $\max_{honest}$ be the maximum verification result for the honest tracks and $\min_{dishonest}$ the minimum verification result for dishonest tracks. We use the "best-evil-to-worst-good ratio"

$$EGR := \min_{dishonest} / \max_{honest}$$

as a performance metric for our simulations. This ratio can be interpreted as follows. If the $EGR \leq 1$, θ_{local} does not exist. Otherwise, there is a *secure interval*

$$\Sigma = (\max_{honest}, \min_{dishonest}) \ ,$$

where any $\theta_{local} \in \Sigma$ (i.e., $\max_{honest} < \theta_{local} < \min_{dishonest}$) results in zero false rejections and zero false acceptances for the 1000 simulated tracks and the given configuration.

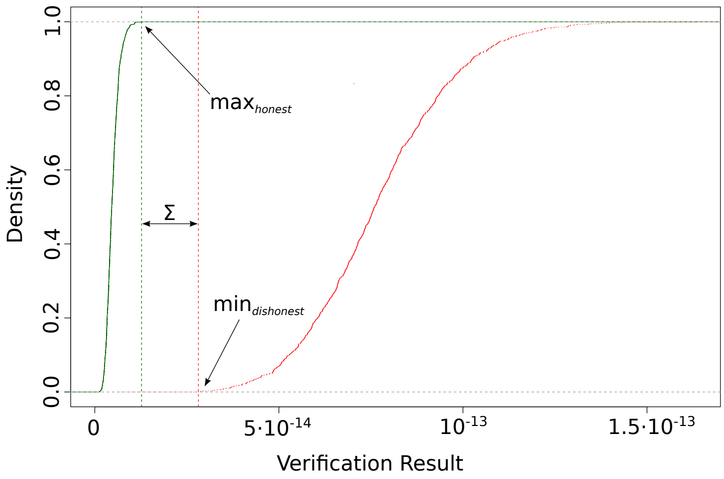


Figure 4.3: Empirical cumulative distribution functions of the verification results for honest (left curve) and dishonest (right curve) tracks with $r = 200$ m, $\sigma = 50$ ns, $\sigma_{drift} = 20$ µs/s, and $|\mathcal{T}| = 19$. Any θ_{local} between max$_{honest}$ and min$_{dishonest}$ (i.e. $\theta_{local} \in \Sigma$) perfectly separates honest and dishonest tracks.

Figure 4.3 shows the empirical cumulative distribution function of the local verification results of an example simulation. In this particular example, we used a radius of 200 m and 19 messages per track. The *EGR* is 2.854879 which is greater than 1 and hence, $\theta_{local} \in \Sigma$ exists. In our results, the estimated secure interval for this configuration is

$$\Sigma = (1.147019 \cdot 10^{-14}, 3.274601 \cdot 10^{-14}) \; .$$

As mentioned above, the verification result directly depends on the simulation radius r. Therefore, we repeated our simulations for different radii. Transferred into a real-world scenario, a greater radius means larger distances between location claims. The results of these simulations are shown in Figure 4.4. For the radii 200 m, 2 km, and 20 km, the *EGR* becomes greater than 1 after a few location claims. In fact, for radii on the order of kilometers, dishonest tracks are perfectly distinguishable from honest tracks after 4 location claims.

If r becomes too small, the optimal θ_{local} does not exist anymore. For instance, the maximum *EGR* for $r = 20$ m is 0.34. That means that there is no θ_{local} which perfectly separates honest and dishonest tracks. This result, however, is as expected since we have chosen a standard deviation for the measurement error which does not enable us to measure such small changes in propagation delay. For $r = 20$ m, the change in distance (and thus propagation delay) for random tracks is on average 10 m, but we have chosen a standard deviation of 15 m for the measurement error.

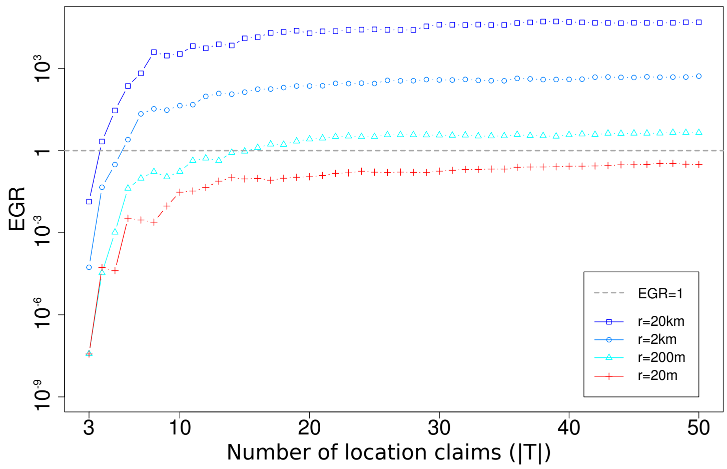


Figure 4.4: Results of simulations for different radii and increasing $|\mathcal{T}|$. In configurations where $EGR > 1$, the optimal threshold θ_{local} exists and our scheme can perfectly separate honest from dishonest tracks.

Figure 4.4 also illustrates that the EGR almost stagnates for more than 15-20 location claims. This knowledge can be used to include a notion of *freshness* into the verification scheme. If an adversary is claiming the correct path in the beginning but lies about its track later on, the verification might work better if only the most recent 15 location claims are considered.

To conclude our simulation results, our local verification scheme only works for tracks on which provers cover distances greater than the system's measurement error. For such tracks, we can say that the greater the distances covered by the prover, the less messages are needed to verify tracks without false acceptances or rejections. To provide a real-world example for distances covered by potential provers, we looked again at data from the OpenSky Network. Airplanes in the en route airspace (i.e., at an altitude of about 30,000 ft) travel at a velocity of up to 300 m/s. That means that they cover distances of the order of kilometers within a few seconds which makes them suitable provers for our track verification scheme.

Results: Global Verification

In case the system has many verifiers covering `R`, our global scheme can be used to reduce false acceptances and rejections for small $|\mathcal{T}|$. To gain insights on the global verification result, we conducted simulations similar to the previous ones. We placed a varying number of verifiers at random positions in a circular area with radius $r = 200$ m and used the same error model parameters (σ and σ_{drift}) as above. We picked this

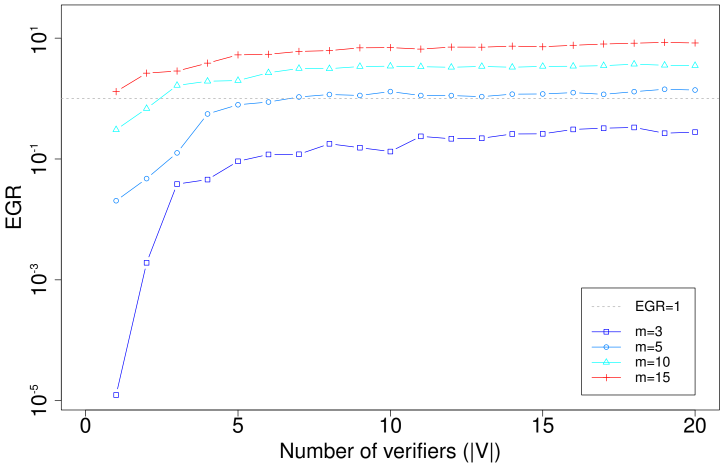


Figure 4.5: Results of the simulation of the global verification scheme for 1000 tracks with an increasing number of verifiers and different numbers of received location claims ($|\mathcal{T}|$). We used the following simulation parameters: $r = 200$ m, $\sigma = 50$ ns, $\sigma_{drift} = 20$ µs/s.

radius because it produces false rejections and acceptances with the local verification for $|\mathcal{T}| \leq 13$ (see Figure 4.4). Compared to larger radii, this is a rather high number of least required messages for doing local verification without false rejections and false acceptances. Thus, there is room for improvement which makes this radius illustrative for the benefits of the global scheme. Furthermore, distances in the order of hundreds of meters are realistic for location claims in aviation.

We ran our simulations for 1000 random tracks to derive $\max_{honest}$ and $\min_{dishonest}$ for each track length $|\mathcal{T}|$ and number of verifiers $|\mathcal{V}|$. Figure 4.5 shows the results. As in the local verification scheme, tracks with just three location claims are still not properly verifiable due to the insufficient number of samples for the noise estimation. However, by increasing the number of verifiers to 7, we can perfectly verify tracks already after the fifth location claim with our global verification scheme. Adding more than 7 verifiers does not result in a significant increase of the EGR.

4.2.4 Realistic Movement Patterns

So far we considered only random movement in order to quantify the effect of input parameters such as the number of verifiers $|\mathcal{V}|$ and the number of location claims $|\mathcal{T}|$ on the accuracy of our verification scheme. However, random movements are not representative since tracks, as we would observe them in the real-world, are usually continuous trajectories. Therefore, we conducted additional simulations with real flight tracks recorded

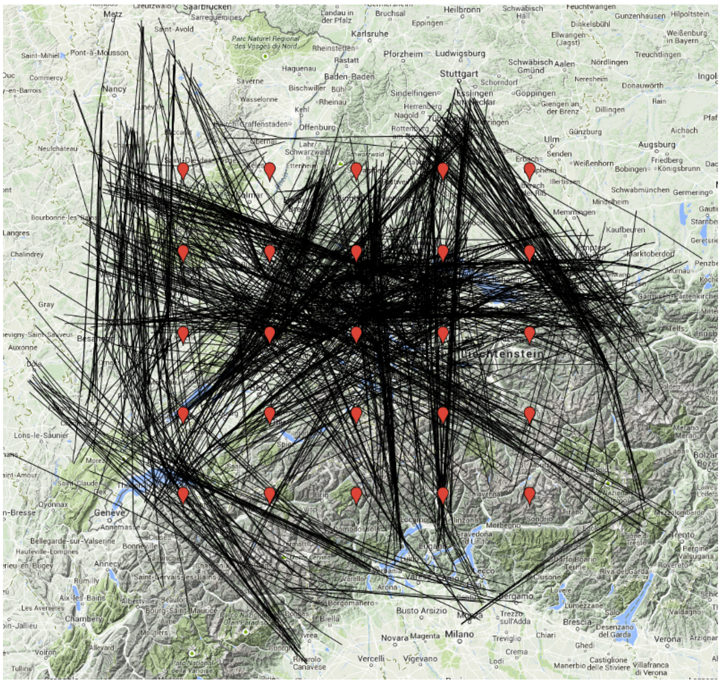


Figure 4.6: The 1000 trajectories fetched from the OpenSky database for our applicability analysis. To achieve a good distribution of verifiers across the considered area, we arranged 25 virtual verifiers on a grid as marked by the bubbles.

by the OpenSky Network. This enables us to better assess the performance of our scheme with realistic tracks, it demonstrates the applicability of our scheme to the ATC scenarios and highlights challenges for secure deployments of our scheme.

Simulation Data and Setup

In ADS-B, airplanes (and other vehicles) periodically broadcast their position, velocity, and other status information. Position reports, for instance, are broadcasted twice per second. They contain the airplane's longitude, latitude, and altitude. Interpreting these position reports as location claims, ADS-B perfectly fits to our track verification scheme.

For our analysis, we fetched 1000 flights recorded by a single receiver from OpenSky's database. To have a large variety of trajectories, we selected a receiver close to Zurich Airport. This way, our simulations contain trajectories from the en route airspace as well as from the approach area of the airport. We placed 25 virtual verifiers in the reception area of the OpenSky receiver. To achieve a good distribution of verifiers across the reception area, we arranged the verifiers on a grid. Figure 4.6 shows the 1000 trajectories and the positions of the 25 verifiers.

While OpenSky provides timestamps with nanosecond precision for the time of arrival of the position reports, ADS-B does unfortunately not support timestamps for the transmission times. An attempt to estimate intertransmission times for two position reports based on the airplane's reported velocity and the distance between the two reported positions failed due to the low resolution of velocity information provided by ADS-B.

It is worth noting that ADS-B also has a feature in which transponders transmit position reports at discrete, known time intervals (see [26, A.1.4.2.3.1]). This allows receivers to estimate the intertransmission times very accurately without the need for explicit transmission timestamps. *Thus, our scheme is fully realizable within the ADS-B standard.* As of this writing, however, the ADS-B deployment is still in an initial phase. Too few airplanes support this mode at the moment and there are no guarantees on the accuracy of the current implementation as it is not yet certified for operational use. Specifically, as of April 2016, data collected with the OpenSky Network (section 5.2) indicates that only 3.5% of ADS-B-equipped aircraft support this mode.

Thus, we had to generate the transmission timestamps $\hat{t}_i$ artificially to be able to apply our track verification scheme to the data. Therefore, we assumed that the track claimed by the airplane was correct and used the verifiers positions to estimate the propagation delay for each position report and verifier. Noise was added to these estimations in the same way as in the previous evaluation (subsection 4.2.3). This way, we were able to apply our scheme to trajectories with realistic properties such as dilution of precision and real shapes.

As in the previous simulations, the timestamps for the position reports were generated with random clock drifts with standard deviation $\sigma_{drift} = 20$ µs/s for each airplane and measurement error with $\sigma = 50$ ns for each timestamp. Then, we calculated the local verification results for all flights for each verifier. In order to gain insights on the time needed to verify a flight, we replayed 50 random position reports of each flight and calculated $\varphi_j(\mathcal{T})$ after the reception of each position report. This approach is meant to simulate a continuous verification of a growing track to imitate a real-world setup.

We fixed the adversary's position to the average position of each flight. As in the above simulations, we assumed for the i-th position report that the adversary uses the same t_i and adjusted the reception times t_i^j according to the differences between the real propagation delay δ_i^j and the expected $\hat{\delta}_i^j$.

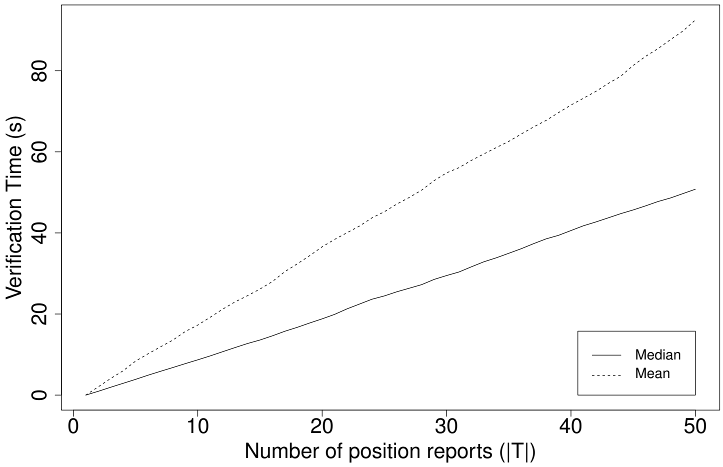


Figure 4.7: Median and mean time needed to receive $|\mathcal{T}|$ position reports. The mean (median) interarrival rate is 0.54 (0.95) messages per second.

Results: Verification Time

The average and median verification time for different $|\mathcal{T}|$ is shown in Figure 4.7. As we will further analyze in chapter 5, the ADS-B communication channel experiences high transmission losses due to noise. This loss results in a lower arrival rate than the transmission rate. According to the standard, the average transmission rate is two position reports per second. However, the average arrival rate observed in the data was 0.54 and the median rate 0.95 messages per second. The difference in mean and median are a result of the high loss close to the edge of the receiver's reception range.

Results: False Alarms/Rejections

The false acceptance rate for $\theta_{local} = \max_{honest}$ and the false rejection rate for $\theta_{local} = \min_{dishonest}$ are shown in Figure 4.8. For example, if we set θ_{local} such that all honest flights get accepted, 2.4% of the dishonest flights get falsely accepted after 15 messages. Conversely, setting θ_{local} such that all dishonest flights get rejected, we observed a false rejection rate of 39.3%. Both, the false acceptance and false rejection rate dropped to zero after receiving 39 position reports.

A deeper analysis of the results revealed that the false rejections and false acceptances are the result of some dishonest trajectories, which produce extremely small $\varphi_j(\mathcal{T})$. The problem is caused by trajectories on which the change in distance to the receiving verifiers is monotonically and linearly increasing or decreasing. If this is the case, the deviation of the adversary's signal arrivals to the expected signal arrivals is also linearly decreasing or increasing. Our clock drift estimator from Equation 4.7, however, assumes that all linear

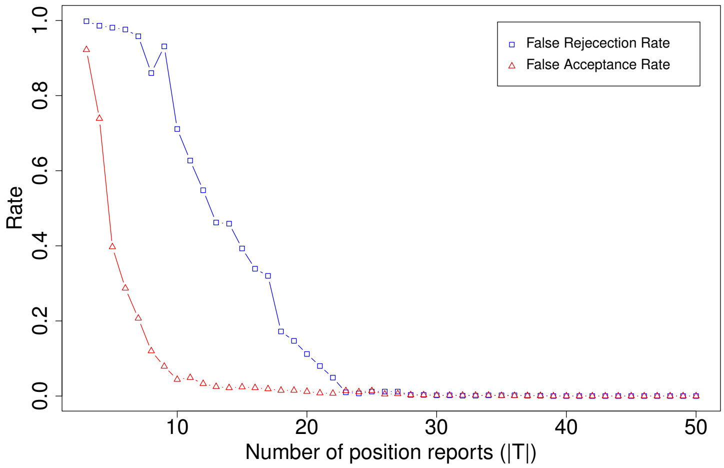


Figure 4.8: False rejection rate for $\theta_{local} = \min_{dishonest}$ and false acceptance rate for $\theta_{local} = \max_{honest}$ of 1000 flights recorded by one receiver of the OpenSky Network.

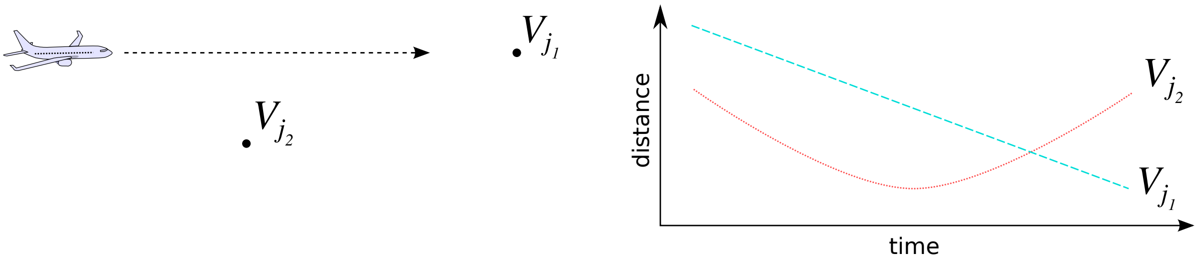


Figure 4.9: Illustrative example scenario with two verifiers and a track which is highly linear from V_{j_1}'s viewpoint and non-linear for V_{j_2}.

deviations are due to clock drift and removes the difference caused by the adversary's dishonesty from the verification result. This results in very small $\varphi_j(\mathcal{T})$ for dishonest tracks $\mathcal{T}$ with a high linearity. Flights, especially en route flights, often have a shape close to a straight line with linearly increasing or decreasing propagation delays. This leads to false acceptances in our simulations. An illustrative example for the linearity property of a trajectory is provided in Figure 4.9. The flight has a high linearity for V_{j_1} and a low linearity for V_{j_2}.

To further investigate this effect, we need a measure for the linearity property of tracks. For this purpose, we quantified the linearity by doing a linear regression over the distances $\|p(t_i) - p_j\|$ for all $m_i \in |\mathcal{T}|$. We then used the root-mean-square error (RMSE) as a measure for non-linearity: the higher the RMSE, the less linear the trajectory from the viewpoint of V_j. Figure 4.10 shows the dependency of the verification results of 1000 flights on the linearity. While the verification result stays constantly small for

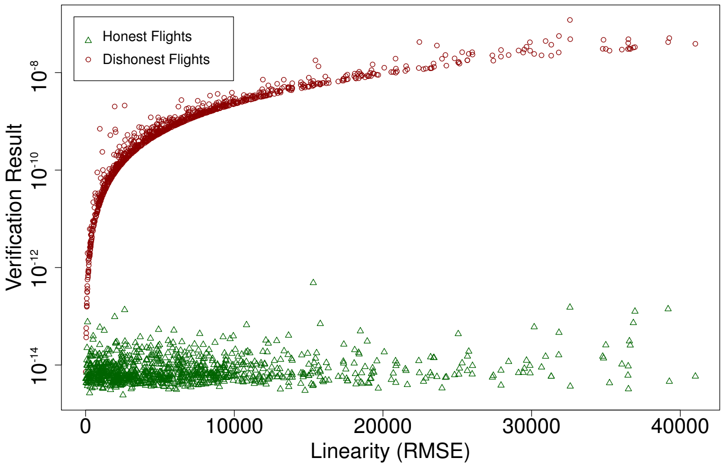


Figure 4.10: The local verification result of 1000 real trajectories consisting of 50 position reports depending on their linearity. Trajectories with a lower linearity have a higher difference between honest and dishonest flights.

honest trajectory claims, the results for dishonest trajectories increases along with the non-linearity.

We also considered the least required RMSE to achieve a zero false acceptance and rejection rate for the local verification scheme. The results are shown in Figure 4.11. We can conclude that the least required linearity becomes smaller the more position reports we used to calculate $\varphi_j(\mathcal{T})$.

As this section showed, it is challenging to deal with linear tracks in combination with clock drift. This might pose a problem for some scenarios. For instance, the least number of required location claims increases if linear tracks are considered. This is problematic in scenarios where fast detection rates are required. To mitigate this problem, we discuss several approaches to avoid or deal with linearity in the following section.

Linearity Mitigation

Long reception ranges result in a high Dilution of Precision (DOP) for tracks far away from the verifiers. Due to DOP, distances between locations further away appear shorter which leaves less room for non-linearity. To tackle this problem, we recommend a *linearity-aware placement* of verifiers in the area of interest. For instance, distributing the available verifiers evenly across the area reduces the distances between tracks and verifiers and thus the DOP. In cases where only certain tracks are possible (e.g. provers are moving on roads or rails), a linearity-aware placement of verifiers can even prevent linear tracks completely.

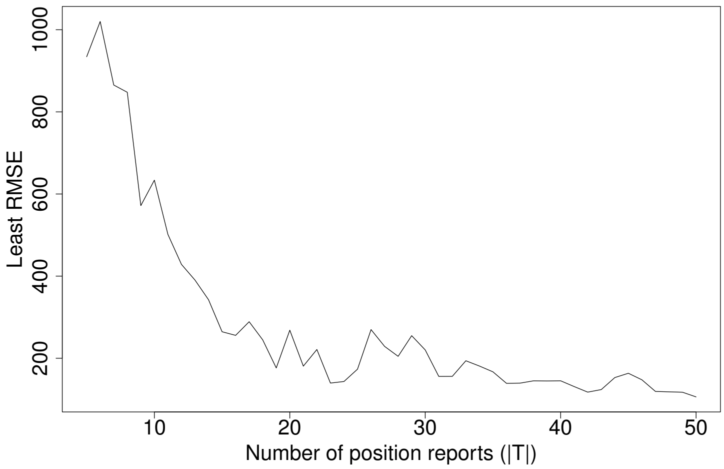


Figure 4.11: The least required root-mean-square error for verifying trajectories with zero false rejections and acceptances.

Another way to deal with linearity is to reduce the least required RMSE by bounding the estimated clock drift. If the upper bound of clock drifts is known and $\hat{t}^j_{drift}$ exceeds it, the track claim might be dishonest. This approach is only applicable if the clock drift bound is lower than the linear deviation caused by the dishonesty of the adversary.

Lastly, a collaborative scheme for estimating $\hat{t}^j_{drift}$ can also prevent attacks which exploit linearity. Therefore, the clock drift coefficients relative to some (or all) of the other verifiers must be determined. This knowledge can then be used to agree on a global $\hat{t}^j_{drift}$, making it impossible to hide linear dishonesty in different clock drift estimations. The pairwise clock drift coefficients can be determined using trusted provers. After a track with a sufficiently high non-linearity has been accepted by the system, they exchange their clock drift estimators and, by that, learn the clock drift coefficients for the other verifiers. In fact, successfully verified provers can even be used to establish a loose synchronization between the provers, making the exchange of drift information unnecessary.

For cases in which none of the above approaches is feasible, verifiers must calculate the RMSE as part of the verification process. Each verifier can then assess whether it is in the position to verify a track or not. Tracks too linear for verification should be ignored.

4.2.5 Discussion

The strength of our track verification scheme lies within its simplicity. Any node which knows its own position and is able to capture the timestamps of the received claims can calculate $\varphi_j(\mathcal{T})$. Besides that, the scheme works completely passively. Except for the track claims, there is no additional communication between verifiers and provers necessary. Hence, verifiers are simple components which can be integrated into existing systems easily. Furthermore, they can run alongside insecure systems such as ADS-B without interfering with their operations.

This simplicity of our scheme enables many applications. In some of them, different threat models might be interesting. Therefore, we discuss several adjustments of our threat model and potential solutions in the remainder of this section.

Mobile Adversary

In subsection 4.2.1, we have proven that our scheme is secure for stationary adversaries. If we remove this assumption, the scheme is not secure anymore. However, the adversary must be able to move in a way such that the propagation delays from the adversaries positions to all verifiers change exactly as they would change if it was on the claimed track. In scenarios where the adversary cannot move freely, for example, due to obstacles in a city, a mobile adversary might not be able to claim arbitrary tracks. Yet, a mobile adversary can be a valid threat in other scenarios. For instance, if the adversary has more degrees of freedom than legitimate nodes, it might be able to successfully claim dishonest tracks. An example would be a vehicular ad hoc network and an adversary that uses a drone to claim dishonest tracks. To prevent such attacks, further security measures need to be in place, such as a reception range sanity check as proposed in section 4.4.

Adversary's Knowledge

Another parameter of the threat model is the adversary's knowledge. In our security analysis, we assumed that the adversary knows everything. In particular, it is aware of its position and the positions of all verifiers. This knowledge makes the verification with $|\mathcal{V}| < 3$ insecure.

Čapkun et al. proposed a scheme for secure location verification where security is based on *covert base stations* (CBS) [34]. By CBS, the authors mean verifiers whose locations are not known to the adversary at the time of execution of the secure location verification. A potential adversary would have to guess the CBSs' positions correctly in order to time the transmissions of its claims without causing inconsistencies at the verifiers. This idea is also applicable to our scheme. The number of verifiers required to securely verify tracks can be reduced with this assumption and it would even be resilient to mobile adversaries. In theory, just a single verifier would be sufficient as long as it is covered. If the adversary is not able to estimate the propagation delay δ_i^j of its signal to verifier V_j, it can only guess the transmission time for m_i and would be detected with high probability.

Finally, the results of chapter 3 suggest that our track verification scheme can also benefit from mobile verifiers. While the mathematical analysis of the security becomes extremely difficult[7], an adversary facing mobile verifiers must keep track of all verifiers to launch exact timing attacks. In fact, mobile verifiers are indeed a realistic scenario. In air traffic monitoring, honest airplanes could act as verifiers. Airplanes equipped with ADS-B receivers and GPS meet all requirements for calculating $\varphi_j(\mathcal{T})$ for surrounding airplanes. If dishonest tracks are detected, the pilot or a on-board system could warn the ground stations. Another advantage of using airplanes as verifiers is that at high altitudes, airplanes can have ranges of more than thousand kilometers[8]. In combination with the high density of today's air spaces, a world-wide coverage could be easily achieved without the need of new infrastructure. In OpenSky, for instance, a single sensor receives position reports of up to 300 airplanes at the same time during peak traffic hours. By being able to use these airplanes for verification, high numbers of verifiers $|\mathcal{V}|$ could be achieved.

Limits Of Our Scheme

As all systems that rely on signal arrival measurements, our scheme is not secure if the adversary is able to transmit independent signals to all verifiers. The adversary could use directional antennas or launch a coordinated attack from different locations. Moser et al. have recently shown in [45] that such attacks are possible using software-defined radios. An adversary capable of such an attack could time the signal arrivals at the verifiers exactly as if they were sent from the claimed positions. However, as mentioned before, such attacks are very sophisticated since they require an extremely accurate timing. In addition to that, the adversary still has to know the exact positions and reception ranges of all verifiers.

---

[7] Each mobile node is adding equations to the system resulting in non-linear and non-quadratic curves.

[8] We assume that communication is possible if there is a line of sight connection.

4.3 Verifying Motion in the Frequency Domain

So far, we have only considered the effects of a prover's mobility in the time domain. Our track verification scheme in effect ignores any velocity information sent by provers. As explained before, the purpose of this ignorance is that it enables a separation of concerns and allows us to provide different methods as independent building blocks. This section investigates how the effects of mobility on wireless transmissions in the frequency domain can be used for instantaneous motion verification. Analogously to the above location and track verification methods based on the MDTOA, the basic principle of our motion verification scheme is to measure an effect of mobility and compare it with the expected value based on the prover's claims. We show experimentally that, in the frequency domain, this basic scheme is not sufficient for real-world ATC communications. Poor frequency accuracy and stability of aircraft transponders demand a frequency synchronization between verifiers to be able to take advantage of the transmission frequency-independent Frequency Difference of Arrival (FDOA). Our measurements show that the extended version of our basic scheme is able to securely verify real-world ATC communications. However, for reasons of clarity and comprehensibility, we start with a Frequency of Arrival (FOA)-based approach first and then extend this basic scheme towards the FDOA.

The basic idea of our scheme is to measure the frequency of a prover's signal with several verifiers at different locations. Each verifier checks the received frequency's conformance with the reported motion claim. As soon as one verifier detects a mismatch, it sends an alarm and we consider an attack detected. Each stationary verifier $V_j \in \mathcal{V}$ located at a known position p_j measures the center frequency $\hat{f}_j$ of the arriving signal carrying a motion claim $m_i = (p(t_i), v(t_i))$. Note that we omitted the transmission timestamp from the claim as it is irrelevant for the frequency domain. The verifier then checks whether the measured frequency $\hat{f}_j$ complies with the frequency expected based on m_i, i.e.,

$$f_j \overset{?}{=} \frac{f_0}{1 - \frac{\|v_j(m_i)\|}{c}}\,, \tag{4.10}$$

where the radial speed $v_j(t_i)$ is calculated according to Equation 2.2 on page 25.

As in the time domain, our basic frequency domain scheme neither relies on any kind of time synchronization among the nodes and it is completely passive and does not require any additional communication. It is particularly efficient in terms of verification speed (considers only single measurement), cost, and scalability. As we will demonstrate in our evaluation, f_j can be measured with simple commercial of-the-shelf hardware and there is no need to, e.g., allocate expensive RF spectrum bandwidth for dedicated verification communication.

4.3.1 Security Analysis

As in our previous analyses, we only consider two-dimensional Cartesian coordinates and vectors. Extending our results to three dimensions is again straightforward. We start our security analysis with the trivial case of a single verifier facing a stationary adversary. Then, we extend the analysis step-by-step by adding more verifiers. We assume that due to the low system requirements of our scheme, cheap hardware enables deployments with large enough numbers of verifiers to achieve a coverage sufficient to provide security. In fact, the following analysis shows that our scheme already provides reasonable security if all positions of interest are covered by at least two verifiers.

Case $\mathcal{V} = \{V_j\}$

In case of a single verifier V_j, a stationary adversary has to imitate the Doppler effect by adapting its actual transmission frequency $\hat{f}_0$ accordingly. Since both verifier and adversary are stationary and thus, $\|v_j(m_i)\| = 0$, the verifier will observe the unchanged transmission frequency, i.e., $\hat{f}_j = \hat{f}_0$. As a consequence, the adversary simply transmits the claim using the frequency expected by V_j according to Equation 4.10, i.e.,

$$\hat{f}_0 = \frac{f_0}{1 - \frac{\|v_j(m_i)\|}{c}} \ .$$

In this way, the adversary can successfully pretend arbitrary motions without being detected by V_j.

This result is the frequency domain counterpart to the previous results for single verifiers in the time domain. We can conclude that, to the best of our knowledge, there is no method to verify positions, tracks, or motions with a single verifier using a omnidirectional antenna and located at a position known to the adversary.

Case $\mathcal{V} = \{V_{j_1}, V_{j_2}\}$

In order to delude two verifiers, the adversary A located at p_A has to find a false motion claim and a transmission frequency which both match V_{j_1}'s and V_{j_2}'s expectations at

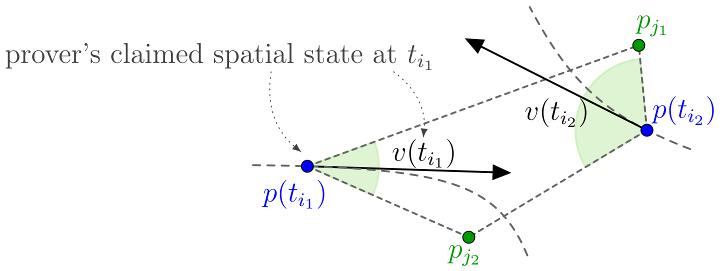


Figure 4.12: Example scenario with two verifiers located at p_{j_1} and p_{j_2} and two motion claims $m_{i_1} = (p(t_{i_1}), v(t_{j_1}))$ and $m_{i_2} = (p(t_{i_2}), v(t_{j_2}))$ which are not verifiable using our scheme. Note that the velocity vectors always bisect the angle between the claimed position and the two verifiers when moving on a hyperbola.

the same time. This means for a specific motion claim m_i, the adversary's transmission frequency $\hat{f}_0$ must satisfy the following two equations:

$$\hat{f}_0 = \frac{f_0}{1 - \frac{\|v_{j_1}(m_i)\|}{c}} \quad \text{and} \quad \hat{f}_0 = \frac{f_0}{1 - \frac{\|v_{j_2}(m_i)\|}{c}} \ .$$

It is easy to see that the two equations can only be satisfied if (and only if) there is a motion claim m_i which satisfies

$$\|v_{j_1}(m_i)\| = \|v_{j_2}(m_i)\| \ .$$

In terms of physical effects, this means that the adversary can only claim motions where the Doppler effect observed by V_{j_1} equals that observed by V_{j_2}. By replacing $\|v_{j_1}(m_i)\|$ and $\|v_{j_2}(m_i)\|$ by their definitions (Equation 2.2), we can reduce the problem to finding a motion claim with

$$\theta_{j_1}(t_i) = \theta_{j_2}(t_i) \ . \tag{4.11}$$

This, in turn, holds only for m_i with $v(t_i)$ bisecting the clock- or counterclockwise angle between the two vectors $(p_{j_1} - p)$ and $(p_{j_2} - p)$. An alternative yet still geometric interpretation of this constraint is that the adversary can only claim motions where $v(t_i)$ is *tangential* to one arm of the hyperbola with focus points p_{j_1} and p_{j_2} and a semi-major axis length of half the difference of the distances from p to the foci. Figure 4.12 illustrates this for two verifiers and two motion claims $m_{i_1}(t_{i_1})$ and $m_{i_2}(t_{i_2})$.

This constraint will already provide sufficient or even strong security for many scenarios since adversaries are forced to claim motion along hyperbolas to remain undetected. Especially in the vehicular network scenario, it is rather unlikely that roads satisfy Equation 4.11. Even if the system faces hyperbolic roads, a proper positioning of the two verifiers can prevent legitimate occurrences of Equation 4.11. In addition to that, if we assume that V_{j_1} and V_{j_2} know each other's positions, they can simply check Equation 4.11 in a first step. In case the equation is satisfied, they should consider the track claim being suspicious. In case it is not satisfied, the verifiers can securely proceed with the normal verification procedure.

Case $\mathcal{V} = \{V_{j_1}, V_{j_2}, V_{j_3}\}$

Analogously to the previous case, we can conclude that an adversary facing three verifiers is limited to motion claims which result in equal Doppler shifts at each verifier, or,

$$\theta_{j_1}(t_i) = \theta_{j_2}(t_i) = \theta_{j_3}(t_i) \ .$$

It is clear that the adversary's options for $p(t_i)$ are now further reduced to positions on one of the extensions of the line segments between each pair of verifiers. In other words, at least two verifiers must lie on a straight line from the perspective of the claimed location $p(t_i)$ in order to satisfy this constraint.

Proof: Let us assume there are no verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ such that a straight line exists which intersects p_{j_1}, p_{j_2}, and $p(t_i)$. This implies that the angles between $v_{j_1}(m_i)$ and $v_{j_2}(m_i)$ are different for all pairs of verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$. We know from the previous case that the adversary has to bisect all these angles in order to satisfy Equation 4.10 for all three verifiers. As it is impossible to bisect two different angles which share one vector simultaneously, any velocity vector will violate Equation 4.10 for at least one verifier.

□

As explained above, an adversary's subsequent motion claims have to comply with their predecessors in order to pass potential sanity checks and remain undetected. Extending the considerations of the previous case to three verifiers leads us to the conclusion that the adversary is now restricted to claiming motion along all three pairwise hyperbolas simultaneously and, thus, the pairwise hyperbolas must be equal. This, however, is only the case if all focus points (the verifiers) lie on one straight line since then, either

$$\theta_x = \theta_y = \theta_z = 0^\circ \quad \text{or} \quad \theta_x = \theta_y = \theta_z = 180^\circ$$

holds.

Conclusion

For $|\mathcal{V}| > 1$, we can conclude from our theoretical analysis in the previous sections that our verification scheme is secure for motion claims $m_i = (p(t_i), v(t_i))$ if there are two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ with different expected radial speeds $\|v_{j_1}(m_i)\| = \|v_{j_2}(m_i)\|$.

It is worth noting that this condition can be guaranteed with an appropriate positioning of verifiers. For instance, covering any location of interest by at least four verifiers in a constellation different from a triangle or a straight line ensures that for every motion claim there is at least one verifier which expects a radial velocity different from the others. The same holds if all locations of interest are enclosed by at least three verifiers. If historical motion claims are considered as well, we can additionally conclude that our scheme is secure if there are two verifiers $V_{j_1}, V_{j_2} \in \mathcal{V}$ and the prover is not claiming to move along a hyperbola with foci p_{j_1} and p_{j_1}.

4.3.2 Evaluation

Real-world measurements are naturally prone to noise induced by hardware and the environment. We are specifically interested in whether we are able to build a real-world system with an accuracy sufficient to reliably detect violations of Equation 4.10. Due to its high velocities and accessibility, we have again selected the scenario of air traffic communication to test the applicability of our scheme in a real-world environment.

ADS-B Environment

The communication medium considered in our evaluation is again the ADS-B. In ADS-B, aircraft broadcast their GNSS-derived location and velocity each about twice per second on the secondary surveillance radar frequency $f_0 = 1090$ MHz [26]. Since the majority of aircraft (e.g. airliners) move on quasi-linear tracks most of the time, we can easily estimate the positions for velocity and velocities for position reports by interpolating the missing information. We then consider both position and velocity reports as a motion claim and obtain an expected rate of about 4 claims per second.

Since the Doppler shift directly depends on the velocity of a target, we are particularly interested in typical speeds of aircraft. Aircraft fly at different speeds depending on the altitude. For instance, the average velocity at altitudes below 9 km is 161 m/s while at higher altitudes, the so called en route airspace, the average velocity is about 230 m/s.

The en route airspace is of special interest to us it is the most crowded airspace. In particular, the data set from the OpenSky Network contained almost twice as many

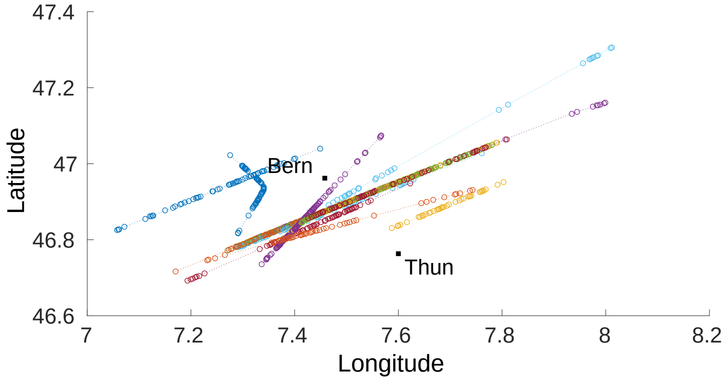


Figure 4.13: Tracks of the 17 flights observed with two USRPs located in Bern and Thun. Each USRP measured the frequency of incoming signals.

positions from the en route airspace than from the lower airspace. That is because aircraft usually climb directly to the en route airspace after take off and only descend to lower altitudes for landing. Hence, we can assume that velocities of about 230 m/s are typical for the air traffic scenario.

A second factor affecting our verification scheme with respect to expected Doppler shifts and number of verifiers required to cover a certain area is the range of a receiver. The transmission power used by aircraft is high enough for reception ranges between 300-400 km if there is a clear line of sight. This enables large scale coverages with a relatively small numbers of receivers.

Experimental Setup

In order to investigate the challenges and accuracy of Doppler shift measurements of ADS-B messages, we deployed two software-defined radios (Ettus USRP X300) at two sites in Switzerland (Bern and Thun) which are about 25 km apart. Both USRPs used a GPS-Disciplined, Oven-controlled Crystal Oscillator (GPSDO) as a clock source and a sample rate of 10 MHz. They provide I/Q samples with a resolution of 14 bit for both in-phase and quadrature component. The radio front-ends (SBX-120 daughterboards) were tuned to the center frequency of ADS-B (1090 MHz). Each setup recorded all ADS-B messages along with their raw I/Q samples using a modified version of the GNU Radio-based software receiver for transponder signals gr-air-modes[9].

After recording ADS-B signals for eight hours, we joined both data sets and identified all messages that were received by both USRPs based on their reception time and transponder ID. We then processed the data with an ADS-B decoder based on the library

---

[9] https://github.com/bistromath/gr-air-modes

provided by the OpenSky Network[10] to extract accurate three-dimensional position, velocity, and heading information from these messages. Figure 4.13 shows the position reports and the locations of the receivers. As a final processing step, we used linear interpolation to estimate the positions for non-position messages as well as velocities for non-velocity messages. With this setup and preprocessing steps, we collected over the course of about 8 hours a set of 2427 ADS-B messages from 17 different aircraft.

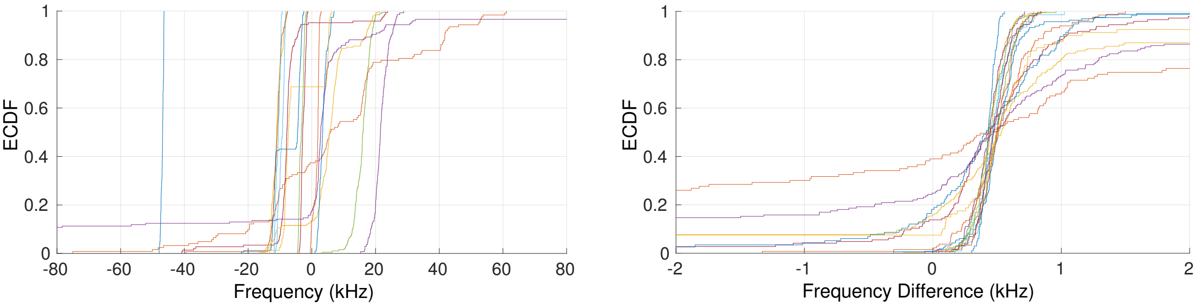
Results: Frequency of Arrival

The first step of our analysis was to investigate the noise level in FOA measurements. A common approach to determine the FOA of a signal is to first translate it to its frequency domain representation using the discrete Fourier transform and then identify the peak frequency. This methodology requires evenly spaced samples of the signal. In ADS-B, however, a pulse position modulation is used and the signal of interest arrives only in short bursts of 0.5 µs with either no, 0.5 µs spacing, or 1 µs spacing (see Figure 5.2 on page 110). Since we are only interested in the peak frequency of these short bursts and not in the plain spectrum, we discard the samples between the pulses to reduce the spectral noise. This filtering results in an incomplete sample set. Unfortunately, algorithms for calculating the spectral representation of a signal based on incomplete samples have generally a higher computational complexity than the Fast Fourier Transform (FFT). The Lomb-Scargle periodigram, as a common approach, scales at $\mathcal{O}(N^2)$ whereas in contrast, the computational cost of the FFT only increases by $\mathcal{O}(N \log N)$ [48]. Hence, for efficiency reasons, we used a linear interpolation to fill the gaps and argue that it is sufficiently accurate since the expected Doppler shift is in the range of a few hundred Hertz while the gaps are at most 1 µs wide. To be more precise, if an aircraft moves directly towards the verifier at a speed of 230 m/s, the Doppler shift is 836 Hz. A gap of 1 µs corresponds to only 0.0836% of a complete oscillation of the Doppler shift frequency and the error caused by the linear interpolation is therefore assumed to be negligible. In summary, our FOA estimation is based on a continuous approximation of the original pulsed signal to amplify the signal's spectral effect.

Let $x_j(k)$ be the k-th I/Q sample of the interpolated signal x_j as received by receiver V_j and $\mathcal{F}\{x_j\}$ its frequency domain representation that results from the FFT. We estimated

---

[10] https://github.com/openskynetwork/java-adsb



(a) Empirical cumulative distribution function of the deviation of the measured FOA of the receiver in Bern from the expected frequency.

(b) Empirical cumulative distribution function of the differences of the FOAs between the two receivers.

Figure 4.14: Results of the frequency of arrival measurements grouped by aircraft.

A/C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
min	302	-109e3	-229	-62	100	171	-35e3	-306	-1328	-18e3	-280e3	130	71	45	-33e3	25	-14e3
mean	434	-868	442	431	441	483	525	733	524	1413	-4117	490	473	479	-139	499	595
median	431	419	454	445	434	464	448	513	504	534	475	494	470	473	416	500	512
max	558	92e3	880	714	960	786	30e3	18e3	1499	18e3	265e3	819	1024	836	2094	844	19e3
std	53	28e3	131	131	134	121	5625	1979	322	8002	59e3	136	152	132	3899	132	5742

Table 4.2: Measurement results: minimum, mean, median, maximum, and standard deviation of the sets of differences $\hat{f}_{j_1} - \hat{f}_{j_2}$ per aircraft between the receiver in Bern (V_{j_1}) and in Thun (V_{j_2}). All values are provided in Hz.

the FOA by determining the peak frequency in the frequency domain representation of the interpolated signal:

$$\hat{f}_j = \arg\max_f |\mathcal{F}\{x_j\}(f)| \ . \tag{4.12}$$

In the next step, we subtracted the expected Doppler shift from $\hat{f}_j$ to obtain the deviation of the measured from the expected FOA. The results are shown in Figure 4.14 and Table 4.2. Figure 4.14a shows the empirical cumulative distribution functions of the FOAs measured by the receiver in Bern and grouped by transponder ID. There are several notable observations.

First, while the messages of some aircraft vary around a central frequency within an interval of less than 10 kHz, others cover the whole spectrum from -300 kHz to 300 kHz. We observed the same patterns for the respective aircraft in the FOAs of the second receiver in Thun. In fact, the outliers of both receivers[11] coincide by 90%. In contrast, the standard deviation of the differences between the two receivers was less than 500 Hz

---

[11] All values above the 10% percentile of the absolute error

for other aircraft (see Table 4.2). We conclude that transponders installed in aircraft differ significantly in frequency accuracy and stability.

Second, the center frequencies of the transponders differ significantly between aircraft as indicated by the horizontal differences in Figure 4.14a. Consequently, we must assume that the actual transmission frequency is different from 1090 MHz as the transponders are obviously not calibrated.

Third, the median difference of the FOAs of all transponders is always close to 500 Hz (see Figure 4.14b). Since this offset is independent of the transponder, we can assume that despite the use of the GPSDO, the radio front-ends of the receivers were not tuned to the exact same center frequency. In addition, Table 4.2 shows that the median difference stays constantly around 450 ± 50 Hz for all flights and the whole duration of our measurements (8h). This suggests that the frequency offset of the two receivers can be assumed to be constant over longer periods of time.

We conclude from our findings that directly applying our verification method (Equation 4.10) to ADS-B is not feasible due to an extremely high noise level. Nevertheless, we can distinguish three kinds of noises that need to be addressed in order to achieve a sufficient accuracy. For this purpose, we propose an adapted version of our verification scheme in the next section. It is specifically tailored, but not limited to dealing with the special conditions of ADS-B signals.

4.3.3 Adapted Verification Scheme

In accordance with the above observations, we can model realistic frequency of arrival measurements with

$$\hat{f}_j = f_0 + \rho_j + \epsilon_j + \epsilon_t + \epsilon' \ ,$$

where ϵ_j is a constant frequency offset of the V_j's radio front-end, ϵ_t is a constant frequency offset of the transmitter, ρ_j the Doppler shift (Equation 2.3 on page 24), and ϵ' a random variable representing measurement noise. Under the assumption that the random measurement noise is additive, the FDOA at two verifiers V_{j_1} and V_{j_2} is then

$$\begin{aligned}\hat{f}_{j_1} - \hat{f}_{j_2} &= (f_0 + \rho_{j_1} + \epsilon_{j_1} + \epsilon_t + \epsilon') - (f_0 + \rho_{j_2} + \epsilon_{j_2} + \epsilon_t + \epsilon') \\ &= \rho_{j_1} - \rho_{j_2} + (\epsilon_{j_1} - \epsilon_{j_2}) + \epsilon \ . \end{aligned} \tag{4.13}$$

Note that the resulting ϵ is a random variable with a variance equal to the sum of both input random variables ϵ'. For the sake of simplicity, we skipped individual noise variables for each receiver.

The idea of our adapted scheme is to verify the FDOA instead of the FOA. As Equation 4.13 shows, this approach has the advantage that the actual transmission frequency ($\hat{f}_0 = f_0 + \epsilon_t$) does not affect the verification. The sources of noise are reduced to the relative frequency offsets of the receivers

$$\epsilon_{j_1,j_2} = \epsilon_{j_1} - \epsilon_{j_2}$$

and the random noise ϵ.

The price for this independence from $\hat{f}_0$ is that the receivers need a high frequency stability and known offset ϵ_{j_1,j_2}. As shown in the previous section, we can in fact assume ϵ_{j_1,j_2} to be constant over a sufficient amount of time. This allows us to learn ϵ_{j_1,j_2} a priori in a calibration phase, for instance by using signals from test transponders which are widely deployed for the calibration of secondary surveillance radar infrastructures. Moreover, as shown by Calvo et al. in [42], modern low cost SDRs already provide extremely high frequency stability with a drift of less than 1 ppm. We therefore assume ϵ_{j_1,j_2} to be known at runtime.

The adapted FDOA-based verification scheme works as follows. For all pairs of verifiers V_{j_1} and V_{j_2} that satisfy one of the requirements derived in our theoretical security analysis in subsection 4.3.1, we check for each received ADS-B message

$$\left|(\hat{f}_{j_1} - \hat{f}_{j_2}) - (\rho_{j_1} - \rho_{j_2}) - \epsilon_{j_1,j_2}\right| \overset{?}{<} \theta_f \ , \tag{4.14}$$

where θ_f is a pre-defined threshold which depends on the measurement error. Note that if there is no measurement error ϵ and offset ϵ_{j_1,j_2}, the left-hand side of Equation 4.14 becomes zero for legitimate motion claims.

Security

As we know from our security analysis, if one of the requirements provided in subsection 4.3.1 is met, the verifiers expect different Doppler shifts, that is $\rho_{j_1} \neq \rho_{j_2}$ for at least two verifiers V_{j_1} and V_{j_2}. Let $\hat{f}_0$ be the real transmission frequency used by an adversary. The FDOA measured by V_{j_1} and V_{j_2} is

$$\begin{aligned} \hat{f}_{j_1} - \hat{f}_{j_2} &= (\hat{f}_0 + \epsilon_{j_1}) - (\hat{f}_0 + \epsilon_{j_2}) + \epsilon \\ &= \epsilon_{j_1,j_2} + \epsilon \ . \end{aligned}$$

Plugging this into Equation 4.14 yields

$$\begin{aligned} \left|(\hat{f}_{j_1} - \hat{f}_{j_2}) - (\rho_{j_1} - \rho_{j_2}) - \epsilon_{j_1,j_2}\right| &= |(\epsilon_{j_1,j_2} + \epsilon) - (\rho_{j_1} - \rho_{j_2}) - \epsilon_{j_1,j_2}| \\ &= |\epsilon - (\rho_{j_1} - \rho_{j_2})| \\ &\overset{?}{<} \theta_f \ . \end{aligned} \tag{4.15}$$

To further evaluate the security of the adapted scheme, we use the false acceptance and false alarm rate as a performance metric. Equation 4.14 and Equation 4.15 show that, generally speaking, false alarms occur when ϵ exceeds θ_f and fake claims become falsely accepted when the absolute difference of ϵ and the expected FDOA is below θ_f. We therefore say that θ_f is optimal if it satisfies

$$\epsilon < \theta_f < |\epsilon - (\rho_{j_1} - \rho_{j_2})| \tag{4.16}$$

for all measurement errors ϵ.

We can conclude that the adapted scheme is secure if θ_f exists. In practice, however, optimal θ_f do often not exist since either the variance of the measurement error is too high or the expected FDOA is not high enough to dominate the measurement error. The scheme can then only provide statistical guarantees based on the distribution of ϵ and the expected FDOA. The next section provides a realistic error model for ϵ based on our real-world measurements.

FDOA Determination

The highest FDOA occurs if an aircraft moves exactly on the line between two verifiers. If we assume a speed of 230 m/s, the difference in radial speed is 460 m/s and the expected FDOA is about 1.6 kHz. Our previous measurements have shown that the accuracy of the above method to determine the FOA is not sufficient. Since the maximum frequency difference exceeded 1.6 kHz (see Table 4.2), an optimal θ_f does not exist. The goal of

A/C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
min	-180	-390	-636	-707	-483	-246	-263	-312	-295	-415	-268	-263	-288	-448	-551	-319	-303
max	117	368	254	222	250	274	241	919	344	375	299	345	379	382	422	294	416
std	52	138	113	100	95	105	69	137	104	103	73	117	116	131	183	112	117

Table 4.3: Frequency-difference of arrival measurement results: minimum, maximum, and standard deviation of the FDOA per aircraft (A/C) minus the constant offset of $\epsilon_{j_1,j_2} = -468.07$ Hz. All values are provided in Hz.

this section is to find a method to determine the FDOA accurately enough for a secure verification of real flights.

A common approach to estimate the FDOA directly from the I/Q samples is determining the offset with the maximum cross-correlation of the frequency domain representations $\mathcal{F}\{x_{j_1}\}(f)$ and $\mathcal{F}\{x_{j_2}\}(f)$ of V_{j_1}'s and V_{j_2}'s signals [49]. This can be efficiently done by applying the convolution theorem:

$$\hat{f}_{j_1} - \hat{f}_{j_2} \quad \approx \quad \arg\max_{f} \left|\mathcal{F}\{x_{j_1}^* x_{j_2}\}(f)\right| \ ,$$

where $x_{j_1}^*$ denotes the complex conjugate of x_{j_1}. In other words, we can estimate the FDOA by calculating the discrete Fourier transform of the product of the conjugated I/Q samples of one signal and the I/Q samples of the other signal.

The drawback of this method is that it requires the transmission of raw signal data to at least one node in the network. While exchanging I/Q samples at a sample rate of 10 MHz appears inefficient at first, we argue that this does not pose a problem for ADS-B messages. More precisely, I/Q samples of USRPs are 32 bit wide[12]. Combined with a sample rate of 10 MHz, we obtain a total of 1200 samples per ADS-B packet. As a matter of fact, only about 580 of these samples actually convey information due to the pulse-based modulation used in ADS-B. Altogether, only 2320 bytes of signal data per verifier and message need to be exchanged to be able to use the above method. More details on network bandwidth requirements are provided at the end of this thesis in subsection 5.3.3.

Results: Measurement Noise

The measured frequency offsets according to this method are shown in Figure 4.15. After removing the estimated Doppler shift in the FDOA measurements, the left-over error is

---

[12] In fact, it is only 28 bit wide since the ADC has an I/Q resolution of 14 bit but we skip this optimization for the sake of simplicity.

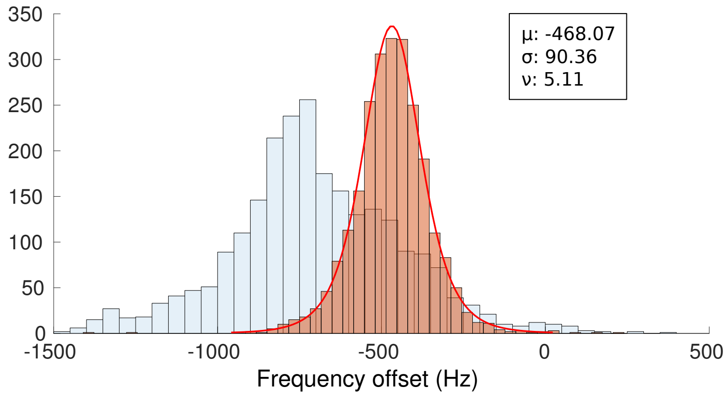


Figure 4.15: Histograms of the frequency-difference of arrival of all messages with (red; narrow) and without (blue; wide) Doppler compensation. The distribution is fitted by a t-location-scale distribution with location μ, scale σ, and shape ν.

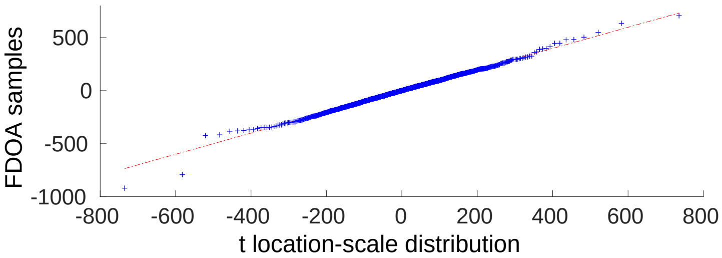


Figure 4.16: Q-Q plot comparing the random measurement error to the fitted t-distribution shown in Figure 4.15 after subtracting the estimated constant tuning offset.

symmetrically distributed around $\epsilon_{j_1,j_2} = -468.07$ Hz. Subtracting the constant offset ϵ_{j_1,j_2} results in a random measurement noise ϵ which is well-fitted by a t-distribution with scaling $\sigma = 90.36$ Hz and $\nu = 5.11$ degrees of freedom (compare Figure 4.16).

The improvement achieved with this method is considerable. In particular, directly measuring the FDOA with the above method turns out to be robust even for highly unstable transponder signals. As the statistics in Table 4.3 show, some transponders are still more stable than others, but the variance dropped significantly compared to the FOA. For example, if we consider the standard deviation as a measure of stability, aircraft 1 still performs best and aircraft 2 worst (compare Table 4.2) while the standard deviation of aircraft 2 is decreased by a factor of over 200. A closer look at the outliers below -345.45 Hz and above 374.8 Hz (99% were between these values) did not reveal any dependence on the signal-to-noise ratio or the bit confidence as defined in [26] and therefore rejects the hypothesis that temporal effects (e.g. higher noise levels) affect some flights more than others. We therefore confirm for our further analysis that the measurement error ϵ is independent and identically distributed for one transponder.

The highest error observed was $\max(|\hat{f}_{j_1} - \hat{f}_{j_2}|) = 918.82$ Hz. In terms of speed, a difference in the FDOA of 918.82 Hz corresponds to a difference in radial speed of 253 m/s. As mentioned above, the maximum difference in radial speed in the air traffic scenario is 460 m/s and the maximum expected FDOA 1.6 kHz. This combined with the result of our security analysis of the adapted scheme (section 4.3.3) suggests, that our setup is not able to perfectly distinguish benign and fake motion claims without false alarms since

$$|\rho_{j_1} - \rho_{j_2}| \leq 2 \cdot \max(|\hat{f}_{j_1} - \hat{f}_{j_2}|)$$

and Equation 4.16 can therefore not be satisfied. However, the probability of such outliers is extremely low. In particular, the probability of a measurement error greater then θ_f is

$$P(|\epsilon| > \theta_f) = 1 - \int_{-\theta_f}^{\theta_f} P(\epsilon = e)\,\mathrm{d}e \ . \tag{4.17}$$

If we assume that the underlying distribution of ϵ matches the fitted t-distribution above (compare Figure 4.16), the probability for errors, e.g, above 538 Hz is

$$P(|\epsilon| > 538 \text{ Hz}) \approx 0.0018 \ .$$

Hence, if we use a threshold $\theta_f = 538$ Hz, the expected false alarm rate is 0.18%. If we further extend our scheme such that an alarm is only raised if Equation 4.14 is violated by two successive motion claims of an aircraft, the average false alarm rate drops quadratically to $P(|\epsilon| > \theta_f)^2 \approx 0.0003\%$.

The same holds for the probability of accepting a false motion claim. A false location claim becomes accepted by the verifiers V_{j_1} and V_{j_2} if $|\epsilon - (\rho_{j_1} - \rho_{j_2})| < \theta_f$. The probability for this inequality to be satisfied can be bounded as follows ($\theta_f \geq 0$):

$$\begin{aligned} P(|\epsilon - (\rho_{j_1} - \rho_{j_2})| < \theta_f) &= P(|(\rho_{j_1} - \rho_{j_2}) - \epsilon| < \theta_f) \\ &\leq P(|\rho_{j_1} - \rho_{j_2}| - |\epsilon| < \theta_f) \\ &= P(|\epsilon| > |\rho_{j_1} - \rho_{j_2}| - \theta_f) \\ &= 1 - \int_{\theta_f - |\rho_{j_1} - \rho_{j_2}|}^{|\rho_{j_1} - \rho_{j_2}| - \theta_f} P(\epsilon = e)\,\mathrm{d}e \ . \end{aligned}$$

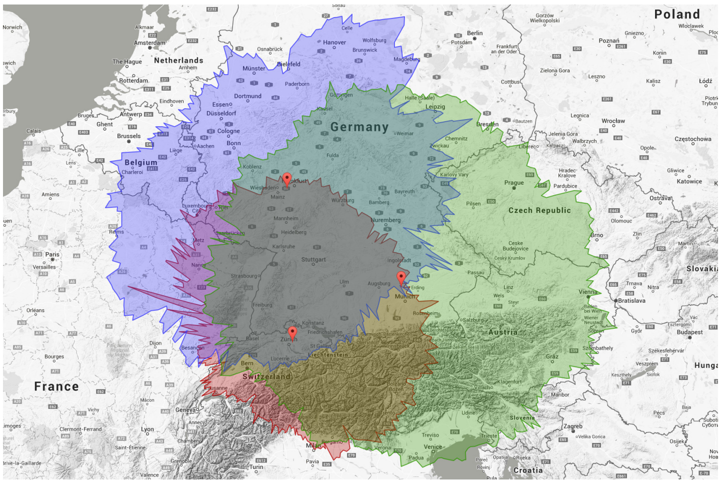


Figure 4.17: Locations and ranges of the three receivers of the OpenSky Network used for our analysis.

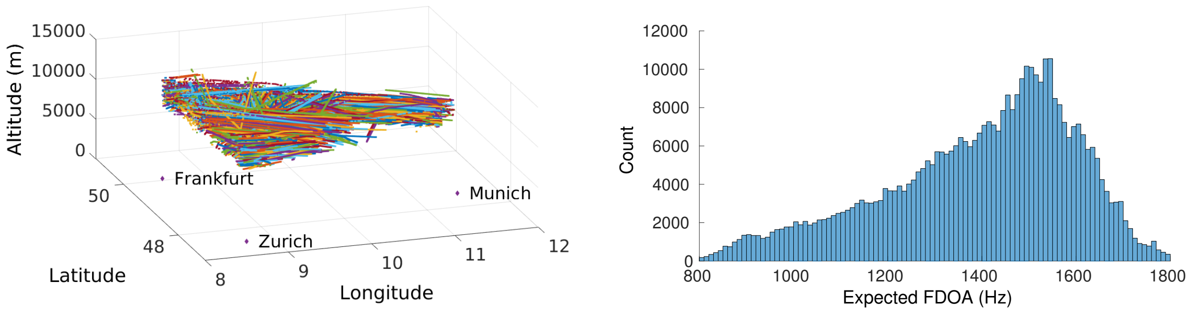
If we assume the expected FDOA of a fake motion claim is 1200 Hz and $\theta_f = 538$ Hz, the probability of a false acceptance is lower than or equal to 0.0676%. If we require the acceptance of two successive motion claims for successful verification, the probability of an adversary not being detected drops to $4.5728 \cdot 10^{-5}\%$.

4.3.4 Real-world Performance

To finally demonstrate that, despite the residual measurement error, our setup is capable of verifying real-world motions with reasonable false alarm/acceptance rates, we conducted simulations based on a large host of real-world air traffic movement data and the above error model. We selected three receivers of the OpenSky Network. Their locations and ranges are shown in Figure 4.17. They are located next to major Central European airports (Frankfurt, Munich, and Zurich) and the triangle between them is fully covered by all three receivers. We have chosen them based on their constellation. The triangle constellation is generally good for our verification scheme as on the one hand, it produces high FDOAs of signals sent from within the triangle, and on the other hand, *there is always a pair of receivers which satisfies the verification requirements* derived in subsection 4.3.1. We then fetched 24 hours (16/03/2016) of state vectors[13] of aircraft flying in the en route airspace (above an altitude of 9 km) within the triangle from OpenSky's database. In total, we obtained 402,627 state vectors from 866 different

---

[13] A state vector consists of aggregated ADS-B position and velocity information over one second from one aircraft. See `https://opensky-network.org` for more information.



(a) The receiver locations and the flights used for our simulations.

(b) The expected frequency-difference of arrivals of according to the positions and velocities reported by the aircraft.

Figure 4.18: The OpenSky 24h data set used for our motion verification feasibility study. In total, the data set consisted of 402,627 position and velocity reports from 866 aircraft.

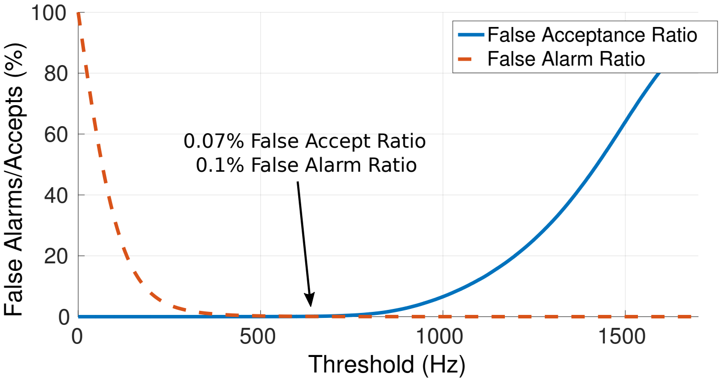


Figure 4.19: The false alarm and false acceptance ratios of our simulation for different thresholds. Each of the 402,627 motion claims was tested individually.

aircraft. The total set of state vectors is shown in Figure 4.18a. We refer to these state vectors as motion claims from here on.

For each motion claim and each pair of receivers, we calculated the expected FDOA. We know from the previous analysis that our verification scheme performs best when the expected FDOA is high. Therefore, we identified for each motion claim the pair of receivers with the highest expected FDOA and used this pair for verification. The distribution of the FDOAs of the receiver pairs with the highest difference are shown in Figure 4.18b. In the next step of our simulation, we added a random measurement error from the t-distribution with $\sigma = 90.36$ Hz and $\nu = 5.11$ to each FDOA value and tested whether it would have been falsely accepted (dishonest prover) or falsely rejected (honest prover) for different thresholds. The results are shown in Figure 4.19. We can

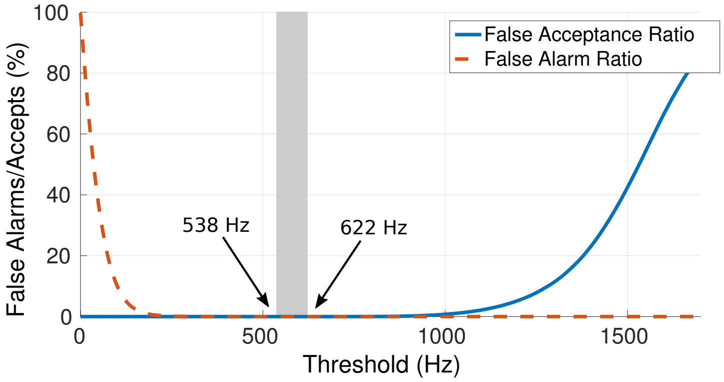


Figure 4.20: The false alarm and false acceptance ratios of our simulation for different thresholds using two successive motion claims. The gray interval marks the thresholds for which both false acceptance and false alarm ratio dropped to zero.

summarize that if the three verifiers perform the verification claim by claim, the best performing threshold would have been $\theta_f = 610$ Hz with a false alarm ratio of 0.1% and a false acceptance ratio of 0.07%. In absolute numbers, 304 of the 402,627 motion claims were falsely accepted and 366 falsely rejected.

We then repeated the simulation but instead of verifying each motion claim individually, we considered two successive motion claims from the same aircraft at a time. In the simulation of an honest prover, we rejected a motion claim if Equation 4.14 was violated by both claims. Conversely, in the simulation of a dishonest prover, we accepted a claim if the claim and its preceding claim both satisfied Equation 4.14. The results for all thresholds are provided in Figure 4.20. Both rates dropped to zero for thresholds $\theta_f \in [538, 622]$. In other words, by using two successive motion claims instead of one, *our scheme detected all attacks while producing no false alarms.*

4.3.5 Discussion

Our previous evaluation has shown that our adapted motion verification scheme is indeed suitable for verifying locations and motions of real-world air traffic. It is notable that ADS-B provides all data already that is required by our scheme to perform the verification. This makes motion verification superior to track verification in the sense that there is no special timing requirement and it is much faster since only two claims are required to achieve a zero false rate as compared to the 39 claims required by track verification. It furthermore verifies not only the location information but also the velocity information.

However, motion verification has several drawbacks that might make track verification preferable in some situations. First, the extraction of I/Q samples requires an SDR-like receiver design which is oftentimes not available in existing hardware. In contrast, timestamps for signal arrivals are usually widely available due to their importance for applications such as multilateration.

Second, in our local track verification scheme, each receiver acts independently from the others whereas in motion verification, the I/Q samples for all received packets need to be collected by a central processing unit to determine the FDOAs. Depending on the number of receivers, the number of aircraft, and the implementation itself this might result in a communication bottleneck. A practical implementation therefore needs to take the scalability into account and an appropriate and careful provisioning of resources is required.

Finally, motion verification requires a frequency synchronization whereas track verification does not require any synchronization at all. However, as we explained above, a loose initial synchronization would suffice since modern oscillators provide sufficient frequency stability. Moreover, due to their wide use in consumer electronics such as smartphones, GNSS receivers have become cheap and could be used by verifiers for frequency synchronization.

4.4 Outlook: Verifying Mobility in the Space Domain

As the previous two sections have demonstrated, both time and frequency domain provide adequate RF channel responses to mobility that are well usable for verifying mobile provers. However, as explained in subsection 2.3.3, the space domain is much more complex, noisy, and volatile. Nevertheless, if verifiers know their vicinity exactly, they can use this knowledge to assess whether the reception of a location or track claim makes sense in terms of a line of sight to the origin of the claim. To be more specific, since a verifier's range is often limited to line of sight communication, exact knowledge about the three-dimensional reception range formed by nearby physical obstacles can be used for a coarse plausibility check. Especially when the prover is mobile, the line of sight connections to a set of verifiers can change often and extremely erratically. This makes it hard for adversaries to comply with the verifier's expectations when transmitting location or track claims while physically being located somewhere else. In the ATC scenario, for example, an adversary, whether stationary, mobile, or distributed, would have to know the exact locations and reception ranges of all verifiers nearby to be able to transmit the fake claims in a plausible manner. If a verifier receives a claim from a location outside its reception range, the claim is likely to be false.

In this section we sketch and discuss a reception area sanity check protocol that might serve as a basis for more advanced space domain plausibility checks. We assume that each verifier $V_j \in \mathcal{V}$ knows its reception area and can check whether a claim m_i is plausible, i.e., was allegedly sent from within the reception range, using a function $\chi_j(m_i)$:

$$\chi_j(m_i) = \begin{cases} 1 & \text{if } p(t_i) \text{ lies within } V_j\text{'s reception range} \\ 0 & \text{else .} \end{cases} \tag{4.18}$$

In an obstacle-free line of sight communication scenario, where the communication is only limited by the free-space path loss, χ_j would be

$$\chi_j(m_i) = \begin{cases} 1 & \text{if } \|p_j - p(t_i)\| \leq r \\ 0 & \text{else} \end{cases}$$

for the maximum reception range r. However, we assume that in practice, reception ranges are much more complex and an initial sampling phase or a more sophisticated propagation model is used to determine χ_j. In fact, as Figure 4.17 on page 98 suggests, the reception range of a stationary receiver is by far not circular but can be learned from past communication.

4.4.1 Sanity Check Protocol

In principle, our protocol simply checks whether the reception of the location claim m_i is plausible or not. Receiving a location claim although the position is not in a verifier's reception range is suspicious and the verifier raises an alarm. In case the reception range covers the claimed location $p(t_i)$, the location claim is accepted and the other verifiers are notified about the reception.

The latter notification is necessary to prevent attacks such as the one depicted in Figure 4.21. Since a single verifier can only detect the presence of a location claim, not its absence, an adversary A could send a location claim for a location within an area of interest `R` only to a single verifier V_{j_1} which accepts it and does not raise an alarm. However, there might be other verifiers which should have received the claim but did not. In order to detect this anomaly, V_{j_1} needs to notify the other verifiers which can then detect the absence of m_i and raise an alarm.

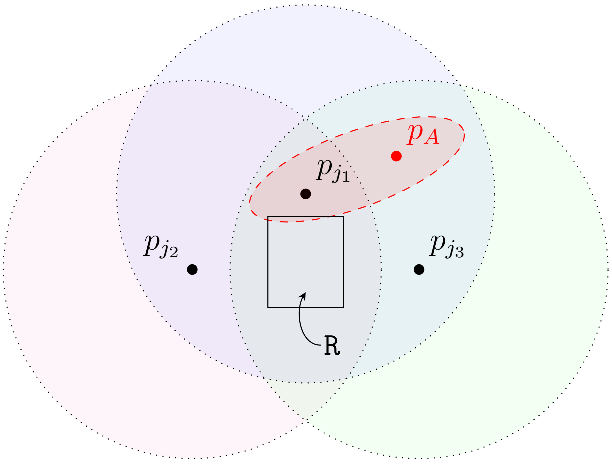


Figure 4.21: Example scenario with three verifiers V_{j_1}, V_{j_2}, and V_{j_3} that are arranged such that the area of interest for `R` is covered by the three verifiers reception ranges (dotted circles). The adversary (located at p_A) transmits its signal to V_{j_1} using a directed antenna (dashed area) and avoids being detected by the other verifiers.

For each reception of a location claim m_i, verifier V_j performs the following verification procedure:

```
if $\chi_j(m_i) = 0$ then // I shouldn't have received this claim
  broadcastAlert($m_i$) // alert all verifiers
else
  if $m_i \notin \mathcal{N}$ then // I received it first
    broadcastNotification($m_i$)// notify all verifiers
  end if
  $\mathcal{M} = \mathcal{M} \cup \{m_i\}$ // add to accepted claims
end if
```

where $\mathcal{N}$ is the set of all received notifications and $\mathcal{M}$ the set of all accepted claims.

The second part of the protocol, which runs in parallel, ensures that location claims are always received by all verifiers that cover the claimed location $p(t_i)$. Assuming that the notification was sent by the verifier with the shortest distance to the prover, all other verifiers should receive the claim at latest after the difference in propagation delays. Let $\epsilon_{\max}$ be an upper bound for the maximum expected measurement error in the time domain. For each received reception notification for m_i from another verifier V_{j_2}, verifier V_{j_1} performs the following procedure:

```
$\mathcal{N} = \mathcal{N} \cup \{m_i\}$ // save notification
if $\chi_j(m_i) = 1$ then // I should also receive this claim
  wait($\delta_i^{j_1} - \delta_i^{j_2} + \epsilon_{\max}$) // wait for it
  if $m_i \notin \mathcal{M}$ then // I should have received it by now
    broadcastAlert($m_i$) // alert all verifiers
  end if
end if
```

The protocol raises an alarm in two cases. The first case occurs when a verifier receives a location claim from a position which is under legitimate conditions not within its reception range. In this case, the prover must be at a position other than the claimed. Second, a verifier does not receive a claim it should have received under normal conditions. Both cases indicate a false claim and attacks such as the one above are detected.

4.4.2 Lossy Communication Channel

Message loss is a natural phenomenon in wireless channels. However, in our protocol above, message loss results in false alarms. Thus, a single alarm does not necessarily indicate an attack and any practical implementation of the sanity check must tolerate some loss. Besides false alarms due to message loss, false alarms due to the legitimate reception of claims from outside the reception range are also possible, although much less likely. Reflections, e.g., from other nodes such as aircraft or the atmosphere can sometimes result in a temporary extension of the reception range. To cope with these two sources of noise, we propose a simple statistical check for tolerating a certain number of alarms.

Let n be the number of transmissions of a track claim or re-transmissions of a location claim and let $\hat{n}$ be the number of false alarms due to channel loss. Under the assumption that loss is a Bernoulli process[14] and the loss probability P_{loss} is known, the expected number of alarms due to channel loss is

$$E(\hat{n}) = |\mathcal{V}| \cdot n \cdot P_{\text{loss}} \ .$$

In addition, since the number of alarms due to channel loss is assumed to be binomially distributed, we can easily build a confidence interval for $\hat{n}$. Thus, a track or location

---

[14] More complex loss models can be used analogously.

claim only passes our statistical sanity check successfully, if $\hat{n}$ is within this confidence interval for a given confidence level α. In other words, if a location claim passes the statistical sanity check, we can be certain with a confidence of α, that the alarms are caused by channel loss, otherwise we assume an attack.

Using the confidence interval check has certain advantages. By choosing an appropriate confidence level, users can control the false positive and false negative detection rate. For instance, higher confidence levels result in wider confidence intervals. On the one hand, this offers adversaries a higher tolerance for its attacks but, on the other hand, the false rejection rate for legitimate tracks will be decreased. In practice, a trade-off needs to be found for concrete application scenarios.

For applications where provers report their location, track, or motion claims over longer periods, another positive side-effect is the behavior of the confidence interval when n increases. Specifically, the confidence interval becomes smaller for each additional location claim at an exponential rate. Figuratively speaking and in terms of security, the sanity check tightens the noose on the adversary with each additional location claim.

4.4.3 Security Considerations

Due to its uncertainty and communication overhead, our space domain sanity check is not a competitive alternative to track and motion verification when it comes to security. However, we argue that putting this simple sanity check aside either of the other verification schemes results in a considerable improvement of their security. The reason is that, so far, we have assumed that all verifiers receive all transmissions since the adversary uses a single omni-directional antenna. Since this does not perfectly reflect the real world and since modern adversaries might use several transmitters or directional antennas, an attack such as the one depicted in Figure 4.21 could be used to avoid a detection by any of the above verification methods. By using this sanity check along with one of the other verification schemes, we force adversaries to send their location claims to all verifiers which cover the spoofed positions. This requirement issues a big challenge for realistic adversaries. In order to launch an attack, they have to know the exact reception ranges of all verifiers and have to be able to control exactly which verifiers receive which location claims. In addition to that, they have to make sure, that the channel loss of their claims is similar to that of honest provers. We argue that achieving such a level of compliance is extremely challenging not only for simple, but also for mobile and distributed adversaries.

I have learned much from my teachers, more from my colleagues, and most from my students.

— Rabbi Chanin in the Babylonian Talmud
(Ta'anis 7a)

5

Air Traffic Surveillance Scenario

Contents

5.1 Automatic Dependent Surveillance-Broadcast 108
  5.1.1 Technical Background 110
  5.1.2 (In-)Security of ADS-B 111
5.2 The OpenSky Network 112
  5.2.1 A Brief History 112
5.3 Challenges for Track & Motion Verification 114
  5.3.1 1090 MHz Frequency Overuse 114
  5.3.2 ADS-B Data Quality 118
  5.3.3 Scalability 121
  5.3.4 Crowdsourcing Security 122

The key insight of the previous chapters is that *mobility improves the security of location awareness in wireless networks*. We have shown that mobility on the receiving side of location information can be used to provide high integrity at low cost. We have further shown that, despite the previous lack of adequate methods, prover mobility itself provides all means required to protect the integrity of spatial information, including velocity and tracks.

In this chapter, we take the next step and investigate the challenges that our new verification schemes have to face in the real world. In accordance with the previous chapter, we use Air Traffic Control (ATC) communication as our test object. There are several reasons for doing so. First, ATC communication is easily accessible since most aircraft are constantly broadcasting many useful information including their spatial state. This makes it a valuable source for real-world traffic data which is a key enabler for realistic and practical research. Second, aircraft are particularly well-suited as mobile provers due to their high velocities and periodic location and velocity broadcasts. Finally, as mentioned in chapter 2 and further elaborated in the next section, the upcoming ATC technology ADS-B has fundamental security issues. These factors combined provide both an interesting research opportunity as well as an important real-world application for our verification schemes.

5.1 Automatic Dependent Surveillance-Broadcast

Today's civil air traffic surveillance is typically based on Secondary Surveillance Radar (SSR). A key characteristic of SSR is that transponders only transmit information upon requests from interrogators. While aircraft can interrogate other aircraft for collision avoidance, the vast majority of interrogations comes from SSRs on the ground. Ground radars consist of rotating antennas which transmit interrogations in a directed beam. Once the aircraft transponder receives an interrogation, it immediately responds with the requested information. By measuring the time between transmission of the request and reception of the reply, the interrogator estimates the distance to the aircraft (ranging). This distance combined with the direction in which the request was sent and the altitude contained in the reply provides the interrogating ground radar with the three dimensional position of the aircraft.

A major drawback of this approach is that update rates for information are limited to the rotation period of the antenna. A full rotation usually lasts about 5-12 seconds. In addition, determining the round-trip time and angle of arrival of an interrogation is

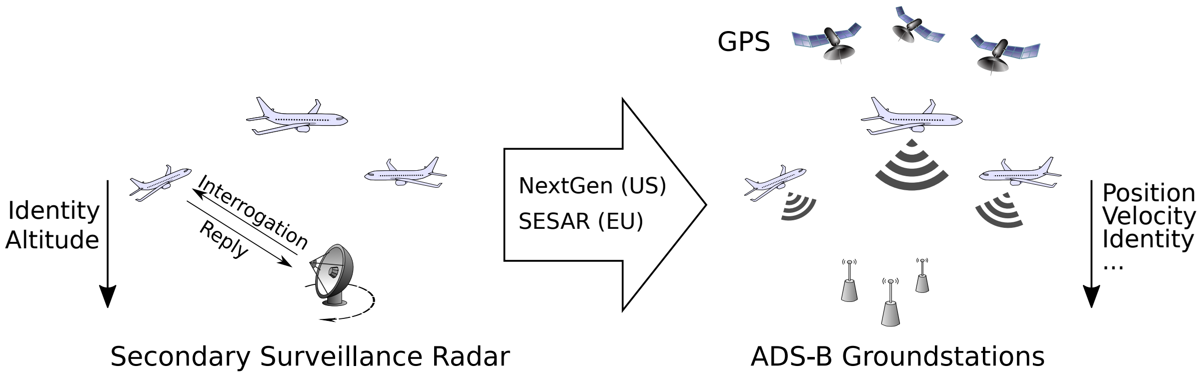


Figure 5.1: The ongoing modernization of civil air traffic surveillance constitutes a switch from ground-based localization to satellite-based on-board positioning systems.

susceptible to measurement errors and precise localization requires expensive techniques such as multi-radar tracking. These shortcomings and the rapid increase in air traffic have led to major modernization programs such as Next Generation Air Transportation System (NextGen) in the US and Single European Sky ATM Research (SESAR) in Europe. A key component of these efforts is the Automatic Dependent Surveillance-Broadcast (ADS-B) protocol. In principle, ADS-B elicits the periodic or event-driven transmission of special SSR transmissions without the need for interrogations. Since ranging is not possible with autonomously transmitted messages, and to achieve a better accuracy, the design of ADS-B requires aircraft to determine their exact locations themselves using satellite-based navigation systems such as GPS. The obtained position and velocity data are then periodically broadcasted over the SSR downlink along with other surveillance information. All receivers that are in line of sight of the aircraft can then simply receive and process the aircraft's spatial state without the need for expensive and inaccurate radars. As ADS-B becomes mandatory in many parts in the world in the late 2010s (Australia[15]) and early 2020s (US[16] and Europe[17]), many airlines have already started updating their fleets with ADS-B capabilities. An overview of the ongoing modernization is provided in Figure 5.1.

---

[15] Instrument number CASA 61/14

[16] Code of Federal Regulations §91.225

[17] Commission Implementing Regulation (EU) No 1028/2014

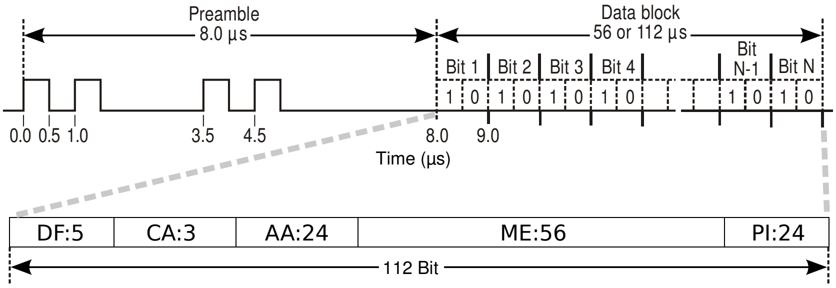


Figure 5.2: Modulation of Mode S replies and the Extended Squitter format according to [52]. The preamble consists of four pulses and is followed by the downlink format, which is set to 17 or 18 to indicate an Extended Squitter. The Extended Squitter provides information about the transponder capabilities (CA) and the unique aircraft address (AA). The 56 bit ME field contains the actual ADS-B data as defined in Appendix A in [53]. The parity identifier (PI) is finally used for error detection.

5.1.1 Technical Background

The ADS-B specification mainly describes the function of broadcasting information [50]. Data link aspects such as the wireless medium or message structures are specified separately and there are two options. The Universal Access Transceiver (UAT) is specifically designed for supporting ADS-B and other aviation services such as Traffic Information Service-Broadcast (TISB) [51]. It is capable of data rates up to 1 Mbps and operates on the 978 MHz RF band. Since UAT requires aircraft to be equipped with costly new hardware, the Federal Aviation Administration (FAA) decided to use UAT only in general aviation[18]. In contrast, scheduled air transportation re-uses existing SSR transponders to broadcast ADS-B. More specifically, they use a general purpose SSR Mode S downlink format which is broadcasted by transponders without interrogation. This downlink format is called Extended Squitter (ES) and the combination of ADS-B and SSR Mode S operating on the 1090 MHz frequency is therefore called 1090ES ADS-B. Due to its higher relevance, we focus on the latter data link for the remainder of this chapter.

Figure 5.2 shows the coding and modulation used in 1090ES ADS-B. Transponders modulate 112 bit onto the carrier frequency (1090 MHz) using a pulse position modulation based on the Manchester code. The 112 bit ES format provides a 56 bit general purpose field (ME) which is filled in by ADS-B with binary coded information. While three-dimensional position and velocity are periodically broadcasted with random inter-transmission times between 0.4 and 0.6 seconds and identity once every 5 seconds, other

---

[18] General aviation refers to all civil flights which do not belong to scheduled air transports.

information such as emergencies and operational status is only broadcasted if triggered by respective events. When operating normally, ADS-B transponders have an average broadcast rate of 4.2 transmissions per second.

5.1.2 (In-)Security of ADS-B

ADS-B has evolved from technologies dating back to World War II, when sophisticated RF technology was not as available as it is today. This led to a negligence of security and ultimately to a lack of security mechanisms in ADS-B. In fact, security has never been a design goal of ADS-B at all. The result of this historical development is that transmissions can be injected, modified or deleted by any attacker who has full control over the wireless channel in a Dolev-Yao [54] manner. With respect to the scope of this thesis, however, we focus on attacks that violate the integrity of spatial information. A more complete overview of attacks, their implications, and potential countermeasures is provided by Strohmeier et al. in [7, 55, 56].

While passive attacks are mainly affecting privacy and might not result in severe risks for air traffic safety, active attacks on ADS-B can result in life-threatening situations caused by misguided pilots, controllers, and avionics. Moreover, advances in wireless technology such as the widespread availability of cheap off-the-shelf software-defined radios have made crafting and transmitting valid ADS-B signals cheap and simple [56]. With no data integrity and origin authentication in place, ADS-B alone is vulnerable to a range of attacks based on transmitting fake transponder signals, including the injection of non-existing (“ghost”) aircraft and the delusion of onboard instruments [57]. A complete list of attack scenarios and physical limitations can be found in [13].

Although these vulnerabilities are known, the long development and certification cycles in aviation make the inclusion of security mechanisms into the ADS-B protocol impossible in the short term[19]. As a consequence, ensuring the integrity of ADS-B-derived information requires passive systems that operate alongside the surveillance infrastructures without interfering with their operations. In addition, the constant cost pressure in aviation demands inexpensive solutions. Low cost was indeed one of the main design goals and selling points of ADS-B and the need for expensive add-ons to provide a minimum level of security would significantly lower its benefits.

We conclude that there is a strong need for passive security measures which do not rely on expensive hardware. The schemes presented in chapter 4 match these require-

---

[19] The development and deployment of new technologies in civil aviation lasts about 20-30 years.

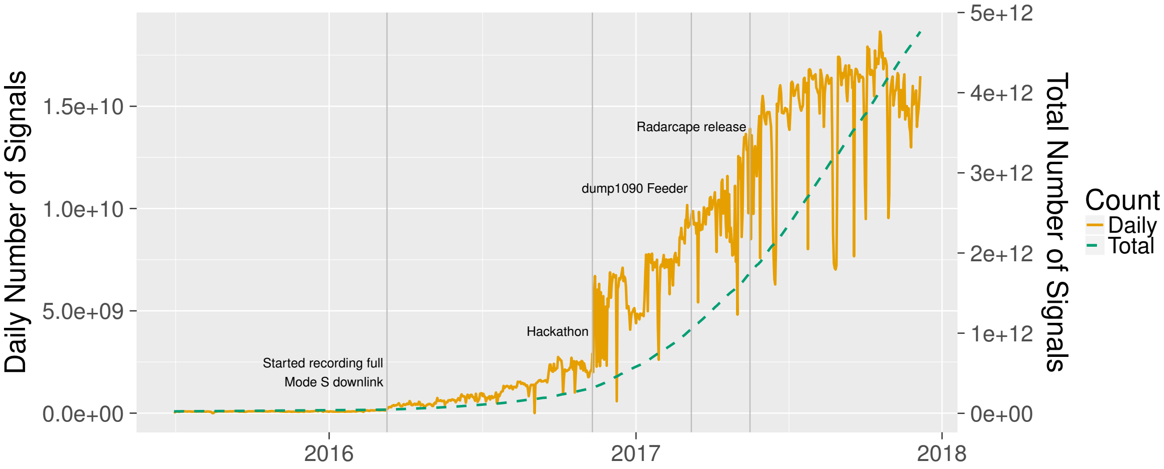


Figure 5.3: The growth of OpenSky's data set between June 2013 and December 2017.

ments. Moreover, they are specifically designed for mobile provers such as aircraft. Implementing them alongside ADS-B would allow the detection of many active attacks which violate the integrity of location and velocity information. However, we have so far only conducted initial measurements and simulations to assess the feasibility given the accuracy of existing hardware and movement patterns observed in air traffic. We therefore dedicate this final chapter to the investigation of additional challenges a real-world implementation of our schemes would have to face in order to secure ADS-B.

5.2 The OpenSky Network

Validation of new technologies and methods is best achieved when enough real-world data is available. In the case of ADS-B, access to large-scale real-world data has only been possible for a few selected industrial and governmental groups in the past. While several live radar visualization services based on ADS-B are available on the Internet, they do not provide the raw data that is most valuable for researchers. For that reason, we have developed the *OpenSky Network*, an open sensor network for research. OpenSky collects and stores all ADS-B and SSR communication captured by sensors that are operated by a worldwide community of volunteers.

5.2.1 A Brief History

We started working on security aspects of ADS-B in 2011. Back then, the availability of large-scale data for analyzing real-world communication behavior of aircraft transpon-

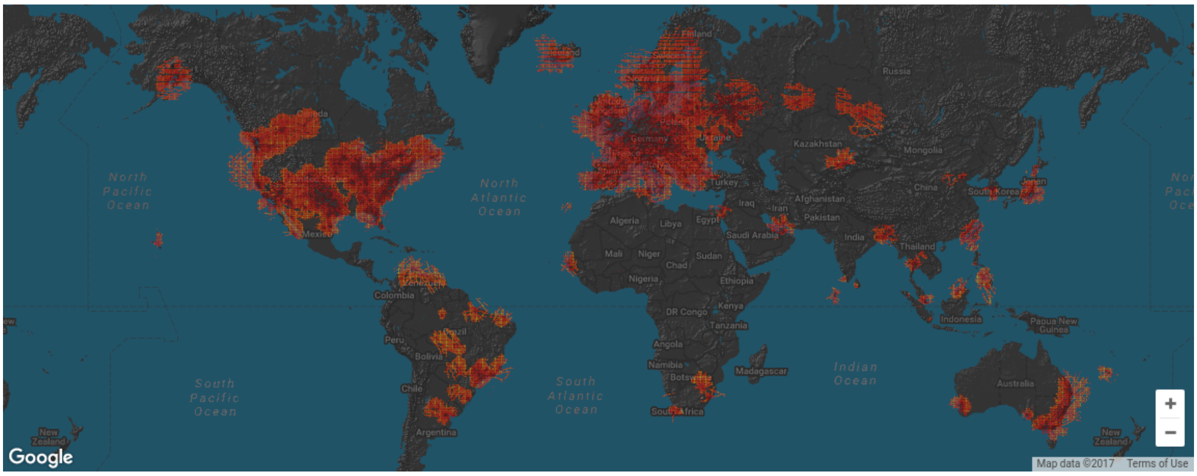


Figure 5.4: Heat map showing the total coverage of the OpenSky Network.

ders was extremely limited. In order to get hands-on experience and capture data for our initial research, we installed three ADS-B sensors in Switzerland. Soon after, we recognized that more data over longer periods was needed not only by us, but also by researchers from other fields. We started installing more sensors, collected the data over the Internet and stored it in a central database. In 2012, the network consisted of 7 sensors deployed in different parts of Switzerland. After dealing with some initial stability problems, we decided to open our network to the public in 2013. In 2014, the network grew further to 15 sensors recording up to 40 million ADS-B transmissions every day. To cope with the enormous growth, we replaced the old three-tier data management architecture with the Lambda architecture proposed by Marz and Warren in [58]. In the meantime, the OpenSky Network Association was founded to provide an open platform for research based on air traffic data. By the end of 2015, the network consisted of 26 sensors and collected 125 million transmissions every day with over 100% year-on-year growth. In May 2016, we extended OpenSky's data set by additionally recording the full SSR Mode S downlink channel. Following the growth trend, OpenSky had 75 sensors collecting 2.7 billion transmissions every day by the end of 2016.

As of this writing (end of 2017), the network size has increased dramatically to more than 1000 sensors recording up to 18 billion SSR Mode S and ADS-B transmissions every day. The growth of the data set over time is depicted in Figure 5.3, the current network coverage is shown in Figure 5.4. More details about the network and its data are provided in [15, 17, 59, 60] and on the project website `https://opensky-network.org`.

5.3 Challenges for Track & Motion Verification

Based on our experiences with the OpenSky Network, we will outline and analyze several challenges that need to be overcome to successfully implement our verification schemes from chapter 4. We identify two different categories of problems that depend on factors such as environment, scale, and available resources. The first category are problems inherent to the ADS-B environment as well as our schemes. Problems that belong to this category are:

- extensive frequency overuse
- poor quality of data provided by ADS-B
- scalability of our schemes

The second category are problems caused by the crowdsourcing paradigm that is used by OpenSky to achieve large scale coverage at reasonable costs:

- required (meta) data not available
- faulty or malicious nodes providing wrong data
- unknown or inaccurate sensor locations

In the following sections, we provide preliminary analyses and propose solutions to these problems. However, most parameters are highly dependent on the environment and the receiver hardware used for the implementation. A representative real-world study is therefore out of the scope of this thesis and we limit our evaluations to theoretical considerations and the information available through OpenSky.

5.3.1 1090 MHz Frequency Overuse

SSR and 1090ES ADS-B are both scarred by the complex organizational and regulatory conditions in aviation. While the development and deployment of new technologies last decades, the safety requirements in aviation combined with the slow adoption of new technologies by key stakeholders such as airlines led to a constant need for legacy compatibility. The latest example is the aforementioned decision to re-use the existing SSR data link for ADS-B. A severe consequence of this legacy driven technological progress is that new technologies inherit weaknesses and limitations of old technologies.

A prime example for this is the 1090 MHz SSR downlink frequency. It was first used by early SSR systems that were operated by military and later civil air traffic surveillance for the identification (Mode A) and localization of aircraft in the late 1950s [61]. The technology rapidly established itself for use in civil air traffic surveillance, especially when altitude reporting functionality (Mode C) was added. However, the system inherited long known limitations from its predecessors. The problems of "fruit" (friendly replies unsynchronized in time) and "garbling" (transponder replies overlapping in time) became increasingly unacceptable as more transponders and ground radars were deployed. This severe limitation ultimately led to the development of SSR Mode S, which is the main means of traffic surveillance in today's ATC infrastructure. In SSR Mode S, the problems of fruit and garbling are mitigated by lower interrogation rates and selective addressing of target transponders during the interrogation. However, a major design goal of Mode S was compatibility with Mode A and C to avoid costly replacement of existing infrastructure. As a result, they share the same up- and downlink frequencies. Moreover, to this day, Mode A and C are still operational in most airspaces in the world since the Mode S equipage is not yet (and might never be) complete. The result of this development is a dramatic RF channel overuse, especially in dense airspaces. In addition to Mode A, C, and S, 1090ES ADS-B now pushes the 1090 MHz frequency band further beyond its capacity limits. Although its rates are much lower and its effects less severe, ADS-B is poised to suffer from this extensive frequency overuse, leading to significant performance degradation in terms of update rates.

This message loss needs to be accounted for by our verification schemes. In the preliminary evaluation in chapter 4, we assumed that all transmissions are received by all sensors equally. This is not true in practice since each sensor location is exposed to a different but potentially overlapping set of transmitters at the same time. As a result, ADS-B transmissions received by one receiver might not be successfully received by another, even though the aircraft was in both receivers' ranges. A quick theoretical analysis shall demonstrate the severeness of this effect.

Let us assume an ADS-B transmission is successfully received if the signal arrival at the receiver does not overlap with the arrival of another transponder signal. For the sake of simplicity, we further assume that every aircraft is equipped with Mode A/C, Mode S, and ADS-B capabilities. Based on the specifications and according to [62, 63], we approximate the average transmission rates and the durations of each transmission of each of the technologies as summarized in Table 5.1. We assigned a type index Θ to each of the technologies for ease of notation.

Technology	Type Θ	Duration Δ_Θ	Average Transmission Rate λ_Θ
Mode A/C	1	20.75 µs	100 Hz
Mode S	2	64 µs	50 Hz
ADS-B	3	120 µs	4.2 Hz

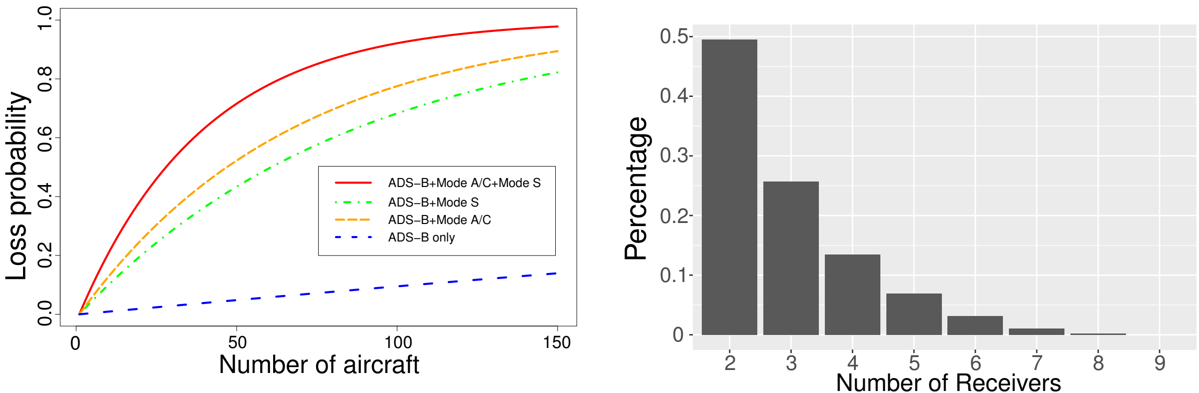
Table 5.1: Assumptions and notations for our probabilistic loss model. The rates for SSR Mode A/C and S are coarse estimates and might significantly differ from airspace to airspace since they highly depend on the availability and configuration of ground infrastructure.

It is important to note that these assumptions are not valid for all airspaces equally. Since most transmissions are elicited by Mode A/C and S interrogations and since these rates highly depend on the availability and configuration of ground infrastructure covering the respective airspace, there is no parameter set which is representative for all airspaces. In effect, the parameters need to be adapted for each receiver location individually. However, our parameter selection is realistic enough to demonstrate the effects of the 1090 MHz RF frequency overuse.

Let t denote the time of arrival of an ADS-B transmission that we aim to receive. According to the assumptions above, we can summarize that a message is lost if a second message of any type Θ with length of Δ_Θ arrives at the receiver within the *vulnerable time interval*

$$\mathcal{I}_\Theta = [t - \Delta_\Theta, t + \Delta_3] \ .$$

Since we consider rather high numbers $|\mathcal{N}|$ of aircraft in the interference range of a receiver, we can assume that the Palm–Khintchine theorem is applicable. As a consequence, although not applicable to single aircraft, the arrivals of the transmissions of all aircraft combined can be modeled as a Poisson process. The probability that at least one message of a type Θ from an aircraft arrives during $\mathcal{I}_\Theta$ is equal to the probability of the complementary event where no message arrives during $\mathcal{I}_\Theta$. Accordingly, we set the number of message arrivals of the process $k = 0$ and the arrival rate $\lambda = \lambda_\Theta \cdot \mathcal{I}_\Theta$. With



(a) The probability for losing ADS-B signals due to interference from other transmitters for different mixes of surveillance technologies.

(b) The distribution of the number of receivers that received ADS-B transmissions. The data set contained only transmissions that were received by more than one receiver and were therefore eligible for our verification schemes.

Figure 5.5: Results of our ADS-B message loss analysis.

n_Θ being the number of message arrivals from a single aircraft for type Θ during $\mathcal{I}_\Theta$ we obtain:

$$\begin{aligned}
P(n_\Theta \geq 1) &= 1 - P(n_\Theta = 0) \\
&= 1 - \frac{\lambda^k}{k!} \cdot e^{-\lambda} \\
&= 1 - \frac{(\lambda_\Theta \cdot \mathcal{I}_\Theta)^0}{0!} \cdot e^{-(\lambda_\Theta \cdot \mathcal{I}_\Theta)} \\
&= 1 - e^{-(\lambda_\Theta \cdot \mathcal{I}_\Theta)} \ .
\end{aligned}$$

Under the assumption that each airplane can only transmit one message (of any type) at a time, we can extend the above probability to the probability that a single aircraft transmits a message. Let n be the number of message arrivals from a certain aircraft which overlap with the ADS-B signal of interest. Then the following holds:

$$\begin{aligned}
P(n \geq 1) &= P(n_3 \geq 1) + \\
&\quad P(n_3 = 0) \cdot P(n_2 \geq 1) + \\
&\quad P(n_3 = 0) \cdot P(n_2 = 0) \cdot P(n_1 \geq 1) \ .
\end{aligned}$$

Hence, the probability that an ADS-B message of one aircraft collides with a transmission of one of the other $|\mathcal{N}| - 1$ aircraft is

$$1 - (1 - P(n \geq 1))^{|\mathcal{N}|-1} .$$

Plugging the rates from Table 5.1 into this simple model yields the collision probability curves shown in Figure 5.5a. It is obvious that the legacy SSR infrastructure interferes extremely with ADS-B. For example, a good[20] receiver setup in OpenSky receives transmissions from over 250 different transponders during peak traffic hours. Under the above assumptions, the probability for an ADS-B transmission to be received without interference from other aircraft is just about 0.17%.

In the current state, our model is overly pessimistic since, in practice, not every overlapping transmission results in a complete loss of information. To get a better idea, we took a one hour data set from the OpenSky Network and analyzed the real-world distribution of number of receivers that received the transmissions. We limited our analysis to ADS-B transmissions that were received by at least two receivers and were therefore eligible for our verification schemes. The results are shown in Figure 5.5b. About half of these ADS-B transmissions were received by only two receivers and the probability decreased following approximately a geometric distribution.

We conclude that our verification schemes have to expect and deal with high loss ratios in practice. Moreover, the high loss probability reduces the rate of usable transmissions significantly with an increasing number of verifiers. Thus, verification times get longer since less transmissions can be used for verification. This problem, however, can be mitigated by including SSR transmissions in the verification. While they do not constitute direct location or motion claims, both values can be interpolated based on preceding and subsequent ADS-B transmissions. In addition, Figure 5.5a suggests that the problem will be extremely mitigated in the long-term if civil aviation manages to perform the complete transition to ADS-B.

5.3.2 ADS-B Data Quality

At the time of writing, ADS-B is still in its deployment phase. Although most airlines have upgraded their fleet with ADS-B-capable transponders, the data accuracy provided varies over a large range. More specifically, the spatial information that is being broadcasted with ADS-B is only as accurate as the underlying onboard positioning sensor.

---

[20] A good setup uses a high gain antenna with a free line of sight in all directions.

SIL	Probability of Exceeding the HCR	Percentage of A/C
0	unknown	24.77%
1	10^{-3} per flight hour or operation	2.89%
2	10^{-5} per flight hour or operation	23.50%
3	10^{-7} per flight hour or operation	46.83%

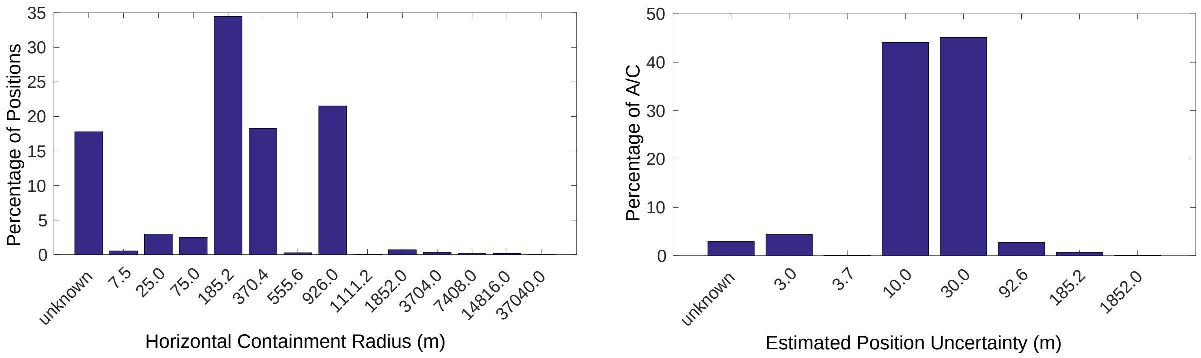
Table 5.2: Surveillance Integrity Level (SIL) reported by aircraft. It indicates the probability of exceeding the horizontal containment radius and only depends on the quality of the position sensor used.

Many aircraft, however, are not equipped with accurate GNSS sensors but rather rely on relatively inaccurate old navigation systems and inertial sensors. While their accuracy levels might still be sufficient for separation purposes, our accurate track and motion verification schemes will produce false alarms if this issue remains disregarded.

The accuracy of ADS-B position reports is indicated by their Navigation Integrity Category (NIC) [50, 2.1.2.12] in conjunction with the Source Integrity Level (SIL) provided separately in operational status reports [50, 2.1.2.15]. The NIC determines the Horizontal Containment Radius (HCR) and the SIL indicates the probability for the actual position to lie within this radius. It is worth noting that the SIL only depends on the on-board sensor used to determine the position. For example, if there are temporarily not enough satellites available for a position fix, the NIC reported becomes 0 (unknown) but the SIL remains unchanged.

In order to prevent false alarms due to inaccurate position and velocity reports, this information needs to be regarded by implementations of our verification schemes. A real-world implementation needs to use individual thresholds for the tolerated deviations in time or frequency based on the reported accuracy level. On the one hand, this lays the system open to attacks where attackers artificially increase the tolerance by falsely reporting unknown or bad accuracy levels. On the other hand, we argue that this will not pose a problem in practice since in aviation, decisions should never be based on spatial information with unknown or bad accuracy. Consequently, we simply recommend to ignore bad data and switch to alternative means of surveillance instead.

To provide more insights into real-world performance of the equipment installed in aircraft, we retrieved and analyzed a large data set from the OpenSky Network. The data set contained all transmissions that were recorded by OpenSky in April, 2016. In total, about 70% of all observed transponders broadcasted ADS-B. The distribution of the HCR reported by those aircraft is shown in Figure 5.6a. In addition, the percentages of aircraft reporting a certain SIL is shown in Table 5.2. About 80.59% of all positions



(a) Distribution of reported horizontal containment radii. The HCR are determined using the navigation integrity category field of each airborne position report.

(b) Distribution of reported estimated position uncertainties (EPU). They provide an upper bound for the error of 95% of the reported positions. The EPU has been determined using the navigation accuracy category field of operational status reports.

Figure 5.6: Results of our ADS-B accuracy analysis.

were reported with an HCR below 1 km. The residual position reports consist of 17.79% with unknown radii and 1.63% with a radius above 1 km.

It is worth noting that a containment radius of 1 km appears large at first but given the hard guarantees associated with it (see Table 5.2), it is rather conservative. To get a more realistic accuracy estimate, we additionally considered the Navigation Accuracy Category (NAC) of the navigation information. Similar to the SIL, it is provided by operational status reports, which are only broadcasted by about 21% of all ADS-B equipped aircraft. The NAC determines the so called Estimated Position Uncertainty (EPU)[21]. The EPU is the 95% accuracy bound for the position fix, i.e., the probability of the distance between the reported and actual position being greater than this bound is 0.05. According to the results shown in Figure 5.6b, the EPU is reported to be at most 30 m in 93.66% of the status reports. This means that the error of positions reported by these ADS-B transponders is at most 30 m in about 89% of the time.

In summary, the accuracy of spatial information provided by ADS-B varies a lot and outliers are to be expected. As a consequence, real-world implementations of our schemes must tolerate these outliers to prevent false alarms. Thresholds need to account for inaccuracies caused by both avionics and the receiving hardware used by verifiers. Moreover, the thresholds need to adapt to the individual accuracy levels reported by

---

[21] In conjunction with GNSS systems such as GPS, the EPU is also called *horizontal figure of merit*.

aircraft. Such an accuracy-aware verification would in effect also verify the upper bounds reported by transponders.

5.3.3 Scalability

In the evaluations of our schemes in chapter 4, we only considered single provers. However, as mentioned earlier in this section, a single receiver in OpenSky can track up to 250 aircraft simultaneously. In case of *local* track verification, this number does not pose a problem. The verification check (Equation 4.1) is very simple and there is no need to exchange any data between verifiers or a central processing unit. This is not the case for global track verification and the FDOA-based motion verification scheme. In both schemes, meta information (timestamps and I/Q samples) need to be gathered centrally to perform the verification. We therefore take a closer look at the network bandwidth requirements in crowded airspaces.

Global Track Verification

In the plain global track verification scheme, local verification results need to be collected by a central server to perform the verification. Well-situated receiver setups in OpenSky track about 250 aircraft at a time. We assume that each aircraft transmits on average 4.2 ADS-B messages per second (see section 5.1). If we ignore the loss issue, a single receiver would receive about 1050 ADS-B transmissions per second. Each verifier has to share at least the ADS-B message content (112 bits) and the local verification result (we assume 64 bit double precision) with the central server to perform the global verification. Hence, without considering protocol overhead, a single verifier produces at least

$$250 \text{ aircraft} \cdot 4.2 \text{ msgs/s} \cdot (112 + 64) \text{ bits/msg} = 184800 \text{ bits/s}$$

of network traffic during peak traffic hours. According to our simulation results (Figure 4.5 on page 75), more than 5 verifiers should be used to keep the number of required messages and thus the verification time low. With 5 verifiers, the data rate increases to about 1 Mbit/s. Note that 5 verifiers is the minimum and in a real-world implementation, higher numbers are more likely. However, even higher numbers of verifiers sum up to data rates of just a few megabits per second which is still well within the capacity bounds of state of the art networks. We conclude that network bandwidth is not an issue in track verification.

Adapted Motion Verification

We have already briefly discussed the problem of network bandwidth in subsection 4.3.3 and came to the conclusion that we need to transfer about 2320 bytes of signal data for each received ADS-B transmission. Plugging this into the equation with 5 verifiers, 250 aircraft per verifier, and 4.2 messages per aircraft per second yields a total data rate of 7.3 megabits per second. While this volume is still manageable, it could easily become an issue for larger setups with tens to hundreds of receivers. For example, the US ADS-B surveillance system consists of about 650 receivers [64]. Under the above assumptions, such a network would produce about 1.5 gigabits of I/Q data per second. However, due to the geographic scale of such a system, it could and should be divided into independent subsystems, each with a manageable network load. The subsystems then perform the verification for just a small fraction of the total airspace and only the verification results or alarms are shared across the whole system.

Besides scaling through a *divide and conquer* approach, appropriate scheduling of the verification process can also lower the system and network load significantly. For example, if re-verification of an aircraft is only required once per minute, the transponder signals received between these verifications do not have to be collected. A subscription-based protocol for exchanging the signal data could therefore reduce the network load significantly. For example, the OpenSky Network tracks about 4000 ADS-B transponders worldwide during peak traffic hours. Let us assume that we want to re-verify each of these 4000 aircraft every minute. To compensate message loss, we collect I/Q data for 20 transmissions from three receivers for each verification. Altogether, this results in a data rate of about 10 megabits per second, which is feasible today even with private Internet connections. We conclude that such a subscription-based approach is suitable to prevent network bottlenecks in motion verification.

5.3.4 Crowdsourcing Security

The previously discussed challenges are all either a result of the ADS-B environment or artifacts of the design of our schemes. For that reason, they need to be addressed in any ATC scenario, regardless of the hardware and software used to implement our schemes. In this final section, we want to consider crowdsourcing as a means of building a large-scale tracking system that has the potential to secure the global air traffic surveillance. In line with the previous section, we focus on challenges that arise from the crowdsourcing paradigm based on our experiences with the OpenSky Network.

Crowdsourcing is a well-established paradigm in the commercial air traffic tracking domain. Volunteers around the world (the "crowd") set up and operate large numbers of receivers and send the tracking data to a central server via the Internet. It has been used by several flight trackers such as Flightradar24[22], FlightAware[23], or PlaneFinder[24] to achieve nearly global coverage within just about a decade. This scale is considerable given that not even aviation authorities have managed to build a globally connected surveillance network. Backed by a large community of aviation enthusiasts, crowdsourcing helps to overcome financial, organizational, and language barriers within a relatively short amount of time. This makes it an interesting candidate for building large-scale networks that can help secure the global air traffic surveillance system in the short term. This immense potential, however, comes at a price. The control over the on-site infrastructure is not with the network operator anymore but with the crowd. This has several implications regarding data availability and integrity which need to be addressed to provide a reliable basis for our verification schemes and other security measures.

It is worth mentioning that, since the OpenSky Network was not specifically built for this application, some of the following problems are not yet solved. As a result, the network does not yet satisfy all requirements that are needed to fully support our schemes. Nevertheless, the network provides us many useful insights into the crowdsourcing paradigm and the challenges associated with it. The remainder of this section is therefore limited to a compilation of the most challenging aspects relevant to our verification schemes. However, as crowdsourcing is not our main focus, a thorough analysis of solutions is out of the scope of this thesis and we will just briefly discuss the solutions at hand.

ADS-B Data Integrity

In a crowdsourced tracking network such as the OpenSky Network, anyone can participate by connecting their receivers to the network. While this fosters growth, it is extremely critical from a security perspective. Attackers can simply connect rogue receivers to the network and inject false information via the Internet without much effort. Examples for potential attackers are skilled Internet trolls, actors on behalf of competing platforms, extortionists aiming for financial gain, or, in the context of this thesis, malicious actors targeting the security services provided by the network.

---

[22] `http://flightradar24.com`

[23] `http://flightaware.com`

[24] `http://planefinder.net`

In order to prevent such attacks, an obvious measure would be to establish some notion of trust to make it harder for attackers to join and interfere with the network. In general, data from a particular sensor can be considered trustworthy if the operator is considered trustworthy. However, the crowdsourcing paradigm prevents the use of classical identity verification methods to build trust in operators. Encryption and authentication are not currently implemented by any of the popular ATC receivers that are used for feeding data flows to the OpenSky Network. Consequently, securing these flows directly would require the use of a non-standard way of feeding, which would ultimately severely hamper the growth of the network. In addition, the success of a crowdsourced network greatly depends on the simplicity of joining the crowd. Complex approaches such as mandatory passport verification of the operator's identity or setting up Pretty Good Privacy (PGP) keys would discourage operators from contributing. For this reason, none of the common methods used in Internet applications to securely authenticate users can be easily applied to the crowdsourcing paradigm.

A more natural method to build trust is through personal long-term relationships and constant communications with other trusted operators and network administrators. Critical applications would then only rely on data from trusted sources. In that way, higher integrity levels can be maintained while the network growth remains unhampered. As a result, attackers have to invest a significant amount of time to become a trusted data provider.

This method, however, cannot be applied to sensors feeding data anonymously to the network. Except for the exchange of sensor data, there is no communication between the network and the operator. As of this writing, about half of OpenSky's receivers are operated by unknown users. Due to the lack of personal contact, building trust in these sensors can only be data-based and trust needs to be re-evaluated constantly. Specifically, anonymous sensors are considered trustworthy if a considerable fraction of their data can be continuously confirmed by existing trusted sensors with overlapping coverages. If the number of mismatches in the data between a trusted and an anonymous sensor exceeds a certain threshold, the anonymous sensor is considered untrustworthy and its data is ignored by the network. Note that anonymous sensors with no common coverage with trusted sensors cannot be verified and their data is therefore considered untrustworthy by default. However, since we have established coverage through known sensors in most parts of the world, the trust of most anonymous sensors in the network can be assessed through the transitivity of this approach.

We conclude that, so far, there is no method to establish *secure* trust while at the same time being perfectly compatible with the crowdsourcing paradigm. Nevertheless, with the above mechanisms in place, potential attackers have to invest much more time and efforts which will discourage at least the spontaneous and less sophisticated attackers. We further argue that only relying on data from sensors of trusted operators provides a similar level of data integrity as maintaining own infrastructure operated by trusted employees.

Inaccurate Receiver Information

All of our approaches are based on the assumption that receiver locations are known. However, in a crowdsourced network such as the OpenSky Network, users might provide inaccurate receiver locations for privacy reasons or out of carelessness. Implementing our verification schemes based on this inaccurate information would result in high false alarm rates. In addition, locations of anonymous receivers (half of the network) are completely unknown, rendering their data unusable for our verification methods.

While the obvious approach would be to simply exclude receivers producing high alarm rates from the verification and ignoring sensors with unknown locations, a far better approach is to accurately locate receivers. To be more specific, being able to locate receivers would not only increase the number of receivers usable for verification. It would also provide means of identification and localization of malicious nodes which would significantly mitigate the problem described in the previous subsection.

One way to locate receivers would be to exploit the reception timestamps for ADS-B transmissions. By assuming that the majority of positions reported with ADS-B are accurate, we can setup a “reverse multilateration” problem which results in accurate receiver locations. More specifically, let $R_1, R_2 \in \mathcal{N}$ be two receivers in the network with overlapping coverage. Let $\mathcal{M}$ be a set of accurate ADS-B position reports received by both receivers. In accordance with OpenSky’s data set, we assume that each position report is delivered by the receivers along with accurate reception timestamps. Then, the receiver positions can be obtained by solving the optimization problem

$$\operatorname*{arg\,min}_{p_{R_1}, p_{R_2}, \tau_{1,2}} \sum_{(p_j, \hat{t}_{j,1}, \hat{t}_{j,2}) \in \mathcal{M}} \left(\hat{t}_{j,1} - \hat{t}_{j,2} - \frac{\|p_j - p_{R_1}\| - \|p_j - p_{R_2}\|}{c} - \tau_{1,2}(\hat{t}_{j,1}) \right)$$

with $t_{j,i}$ being the reception timestamp of R_i for the j-th position report and $\tau_{1,2}$ being a function modeling the clock offset of both receivers over time, i.e., their relative clock drift.

Two receivers located close to each other typically receive several millions of ADS-B position reports per day from large numbers of different aircraft. We therefore argue that there is enough data available to solve this optimization problem accurately. Moreover, if we assume that at least some receiver locations are known and reliable, we can use this method to verify untrusted receiver locations. In fact, we have conducted preliminary tests with synchronized receivers (i.e, $\tau_{i_1,i_2}(t) = k$ with constant k for all t) and our results indicate that receiver locations can be determined with a horizontal accuracy of less than 10 meters compared to the GPS location reported by the devices. In simulations, we even obtained a horizontal accuracy of less than 1 meter and a vertical accuracy of less then 2 meters. However, although our tests prove the feasibility of this approach, more research on clock synchronization, geometrical errors, and the accuracy of positions provided by ADS-B is required to better assess real-world performance.

Data Availability

The final challenge is the availability of receivers operated by volunteers with varying levels of skills, motivation, financial resources, and infrastructure. While most volunteers provide a stable 24/7 Internet connection and a constant power supply, others turn off their receivers during night or do not properly monitor the online status of their device. As a result, there is a considerable fluctuation in the network which might be intolerable when it comes to providing security guarantees.

As of November 2017, the OpenSky Network has constantly between 700 and 800 receivers online, while over 250 registered receivers are offline. The majority of these offline receivers were never online or have been offline for more than 3 months. This suggests that some users register non-existing receivers, e.g., to test our platform while others stop feeding data to our network for unknown reasons. However, those receivers that are online have in fact a rather high availability. Over 80% of receivers seen during a period of one month have an availability of 100%.

We conclude that the majority of volunteers seem to have an intrinsic motivation to operate their receivers reliably. To improve the overall availability in the network, many tracking platforms have rewarding schemes in place. For example, commercial trackers usually reward their feeders with free business subscriptions which include extended features and access to additional information. Besides such rewards, gamification through receiver rankings further increases the motivation of volunteers to optimize the performance of their setups. Nevertheless, it remains hard to provide guarantees regarding availability and coverage for certain areas. In densely populated areas with many volunteers (e.g., Central Europe), single offline receivers do not affect the network's coverage. Receiver outages in less populated or less wealthy areas of the world, however, may create blind spots due to the lack of redundancy.

Besides the availability of receivers, which determines the fundamental availability of data, different receivers provide different levels of information. This results in an inconsistent coverage of the meta data required by our verification schemes. The OpenSky Network currently supports four different types of receiving hardware and protocols:

- Kinetic Avionics' SBS-3,
- Günter Köllner's Radarcape,
- the open-source software receiver dump1090, and
- any receiver supporting the ASTERIX CAT21 protocol.

The Radarcape, for example, provides GPS-synchronized nanosecond timestamps for signal arrivals which enable track verification. In contrast, the ASTERIX CAT21 protocol does not provide any timestamps at all, rendering any time domain verification infeasible. Since volunteers can choose the hardware at will, the availability of meta data is not under the control of the network. If the network wants accurate timestamps for a certain region, it needs to provide infrastructure (receivers) and find hosts. As for motion verification, none of the above receivers provide the signal data required to determine the FDOA. In this case, new hardware or software for existing SDRs is required to perform motion verification.

In summary, availability and accuracy guarantees are difficult to provide based on crowdsourcing. Furthermore, establishing coverage and obtaining the required meta data needed by our verification schemes still requires significant investments and efforts by the network operator. Since security applications typically demand high accuracy and availability, crowdsourcing alone is not the perfect solution. However, as there are currently no security measures in place at all, any implementation, unreliable or not, would constitute a significant improvement. Moreover, "fuzzy" statistical plausibility checks such as those proposed by Strohmeier et al. in [20] could still be applied as they simply rely on decoded tracking information.

Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.

I have made this longer than usual because I did not have the time to make it shorter.

— Blaise Pascal in *Provincial Letters: Letter XVI*

6

Summary & Future Work

Contents

6.1 Key Results 130
6.2 Future Directions 132
6.3 Final Conclusions 132

6.1 Key Results

We have demonstrated that mobility in wireless networks improves the security of location awareness in several ways. Prior to this thesis, research has only focused on solving the problem of *location* verification which aims, by definition, at verifying the accuracy of location information provided by *stationary* nodes. Although there exists a considerable body of work on this problem, all proposed solutions suffer from at least one of following flaws:

- requires time synchronization
- requires extra communication
- requires dedicated hardware
- is only secure against blind attackers
- requires many verifier nodes
- poor spacial resolution
- can only provide “soft” security guarantees

Our literature review on location verification (compare Table 2.2 on page 21) has shown that existing methods mainly trade some of these requirements for others.

In contrast, we have provided a new solution to location verification in chapter 3 which exploits the mobility of verifiers to overcome all of the aforementioned flaws. Specifically, by using moving instead of stationary verifiers, we have shown that our scheme is able to provide provable security in a two-dimensional setting with only two passive verifiers. By measuring the time domain effects of their mobility on wireless transmissions of claiming nodes and by adapting their movement to the claimed positions, two verifiers are enough to detect inaccurate location claims after two transmissions. These results demonstrate that by adding mobility as an ingredient to location verification, system requirements can be significantly reduced, while at the same time hard security guarantees can be provided.

Another result of our literature review is that researchers have largely ignored the case of verifying moving nodes in the past. As a result, there are no adequate methods available to deal with mobile provers. Most existing verification schemes are practically unusable if the node of interest changes its location during the verification process. To fill this gap, we have proposed two more methods that are specifically designed for mobile provers. We identified the two new problems of track and motion verification and proposed solutions to each of them. On the one hand, our solution to track verification verifies sequences of locations ("tracks") based on the mobility-differentiated time of arrival of the prover's location broadcasts. Our analysis and evaluation has shown that our scheme is not only secure in theory, but is also able to securely verify real trajectories of aircraft with off-the-shelf hardware. On the other hand, our motion verification scheme exploits the Doppler effect of transmissions of a moving node to securely verify its position, direction of movement, and speed at the same time. Our experiments with real aircraft signals have shown that compared to track verification, this is approach is much faster (instantaneous verification) and more accurate as it also includes velocity information. However, motion verification is also more expensive in terms of system requirements as it requires a specific receiver design, frequency synchronization, and the exchange of signal data to measure the Doppler effect. We therefore conclude that, depending on the available resources, motion verification is preferable in terms of accuracy while track verification is cheaper and easier to implement. Both methods, however, demonstrate that verification of moving provers can be secure, simple, and cheap. Overall, our results in location, track, and motion verification prove that mobility, whether on the verifying or on the claiming side of the network, provides means to ensure high integrity of location information with light-weight methods.

To provide more insights on the verification of mobile nodes in a real-world setting, we conducted additional analyses of conditions in the air traffic surveillance scenario in chapter 5. While we have already proven the technical feasibility of our schemes with state-of-the-art hardware in chapter 4, the objective of these additional analyses was to identify further challenges that are to be expected in a real-world setting. To reach this research goal, we have built a crowdsourced sensor network to collect air traffic data for research purposes. The OpenSky Network has grown tremendously since its launch in 2013 and has achieved coverage in all parts of the world. The large-scale data collected by the network has helped to produce the results presented in this thesis. Moreover, the network continues its operation as a data provider for various research fields, including but not limited to aviation security research. It also continues to monitor the ADS-B environment to identify potential threats to the safety of the air transportation system as early as possible. As for the real-world applicability of our schemes, we found that in the case of ADS-B, individual thresholds for different aircraft should be applied to prevent high false alarm rates due to differing equipage and accuracy levels. Furthermore, ADS-B suffers from an extreme frequency overuse and the high probability of message loss is likely to increase verification times in practice.

6.2 Future Directions

Based on our insights during the course of this thesis, we identify two research directions which have the potential to further improve the security of location awareness and provide long-term support for secure intelligent transportation systems.

Stronger threat models: Although researchers have demonstrated their feasibility, we still rate sophisticated attacks on location aware networks with distributed or mobile adversaries rather unlikely and difficult with state-of-the-art devices. However, the example of ADS-B security has taught us to prepare for stronger threat models as early as possible. One important research direction therefore is a more thorough investigation of the feasibility of and defense methods against such stronger adversaries, especially in the presence of mobility. With the ongoing proliferation of mobile platforms such as drones and with our transportation systems becoming increasingly intelligent and autonomous, the need for even more robust schemes becomes more and more pressing.

Security and crowdsourcing: As we have mentioned in chapter 5, the state of the art in crowdsourcing air traffic communications is not sufficient to provide an adequate basis for our verification techniques. However, crowdsourcing is extremely powerful due to its enormous potential growth. To ultimately leverage this growth for security (or other applications), the analysis and the development of methods to better control the infrastructure provided by the crowd are required. While cheap yet good hardware is already available, significant efforts need to be invested in software development to integrate the methods into widely used software. Additionally, online marketing and communication channels to the community need to be improved to distribute the right software to the crowd and to scale in terms of organizational overhead. The methods and experiences gained throughout this process can also help building similar projects for other domains.

6.3 Final Conclusions

While most of the work presented in this thesis was clearly motivated by the air traffic scenario, the reader should keep in mind that our schemes are not limited to this application. Although the accuracy of currently available RF receivers might not be sufficient to measure the MDTOA or FDOA of a prover moving at walking speed, the rapid technological progress might allow us to detect even the slightest location changes in the future. In addition to that, a viable yet limited alternative to radio communication is the ultrasound medium. Due to the low propagation speed of sound waves, both the MDTOA as well as the FDOA are much more pronounced than in RF communications. A simple experiment with laptop speakers and built-in laptop microphones allowed us to measure the received frequency of ultrasound signals with an average accuracy of less than 1 Hz. In terms of Doppler shift, 1 Hz corresponds to a radial velocity of about 2 centimeters per second. While this is a significant improvement in accuracy, ultrasound communication is much more limited in range and more susceptible to environmental noise.

On a final note, the public reactions of aviation authorities and organizations on the security issues of ADS-B have switched during the course of our research from denial[25] to recognition[26] and finally to action[27]. However, authorities will continue to deploy ADS-B along with all its security issues and there is still a long way to go (not only by the aviation sector) to implement sustainable protection of today's and future intelligent transportation systems against the increasing capabilities of attackers. We believe that our results constitute a considerable step in this direction. Nevertheless, a close collaboration between researchers, industry, authorities, and other stakeholders is needed to smooth the way into a secure and efficient future of transportation. The case of ADS-B should teach us that it is better to be secure from the beginning than sorry in the end.

---

[25] In 2010, the FAA stated that "*using ADS-B data does not subject an aircraft to any increased risk compared to the risk that is experienced today*" [65].

[26] In 2013, after a workshop on CNS security, the NATO/EUROCONTROL ATM Security Coordination Group released a security statement indicating that the security risks associated with ADS-B are not fully understood [66].

[27] In 2014, the International Civil Aviation Organization (ICAO) released a set of recommendations to its member states for actions to approach the security issues associated with ADS-B [67].

Bibliography

[1] A. Newell and H. A. Simon, “Computer science as empirical inquiry: Symbols and search,” *Communications of the ACM*, vol. 19, no. 3, Mar. 1976.

[2] J. Zhang, F. Y. Wang, K. Wang, W. H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, Dec. 2011.

[3] P. Varaiya, “Smart cars on smart roads: problems of control,” *IEEE Transactions on Automatic Control*, vol. 38, no. 2, Feb. 1993.

[4] National Research Council, *Autonomy Research for Civil Aviation: Toward a New Era of Flight*. The National Academies Press, 2014.

[5] O. Levander, “Forget autonomous cars–autonomous ships are almost here,” *IEEE Spectrum*, Jan. 2017.

[6] A. Thompson, “Self-driving freight trains are now traveling the rails without a human on board,” *Popular Mechanics*, Oct. 2017.

[7] M. Strohmeier, V. Lenders, and I. Martinovic, “On the security of the automatic dependent surveillance-broadcast protocol,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, Oct. 2015.

[8] B. Parno and A. Perrig, “Challenges in securing vehicular networks,” in *Proceedings of the workshop on hot topics in networks (HotNets-IV)*, 2005.

[9] M. Raya and J.-P. Hubaux, “Securing vehicular ad hoc networks,” *Journal of Computer Security*, vol. 15, no. 1, Jan. 2007.

[10] B. DeBruhl, S. Weerakkody, B. Sinopoli, and P. Tague, “Is your commute driving you crazy?: A study of misbehavior in vehicular platoons,” in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, Jun. 2015.

[11] Andrei Costin and Aurélien Francillon, “Ghost is in the Air(traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices,” Black Hat USA, Jul. 2012.

[12] Brad Haines, "Hacker + Airplanes = No Good Can Come Of This," DEF CON®20 Hacking Conference, Jul. 2012.

[13] M. Schäfer, V. Lenders, and I. Martinovic, "Experimental Analysis of Attacks on Next Generation Air Traffic Communication," in *Proceedings of the 11th International Conference on Applied Cryptography and Network Security (ACNS)*. Springer Berlin Heidelberg, 2013.

[14] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B," *IEEE Communications Magazine*, 2014.

[15] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing Up OpenSky: A Large-scale ADS-B Sensor Network for Research," in *Proceedings of the 13th ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN)*, 2014.

[16] ——, "Demonstration abstract: Opensky: A large-scale ads-b sensor network for research," in *Proceedings of the 13th ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN)*, 2014.

[17] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, R. Pinheiro, V. Lenders, and I. Martinovic, "OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage," in *Proceedings of the 35th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2016.

[18] M. Schäfer, V. Lenders, and J. B. Schmitt, "Secure Track Verification," in *Proceedings of the 36th IEEE Symposium on Security and Privacy (S&P)*, 2015.

[19] M. Schäfer, P. Leu, V. Lenders, and J. Schmitt, "Secure Motion Verification using the Doppler Effect," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, Jul. 2016.

[20] M. Strohmeier, M. Smith, M. Schäfer, V. Lenders, and I. Martinovic, "Crowdsourcing Security for Wireless Air Traffic Communications," in *Proceedings of the 9th International Conference on Cyber Conflict (CyCon)*, May 2017.

[21] Y. Zeng, J. Cao, J. Hong, S. Zhang, and L. Xie, "Secure localization and location verification in wireless sensor networks: a survey," *The Journal of Supercomputing*, vol. 64, no. 3, Jun. 2013.

[22] *Commission Regulation (EU) 2016/919*, European Commission, May 2016.

[23] M. Hartong, R. Goel, and D. Wijesekera, "Securing positive train control systems," in *IFIP International Conference on Critical Infrastructure Protection (ICCIP)*. Springer US, 2008.

[24] I. T. Union, *Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band*, Feb. 2014.

[25] M. Balduzzi, A. Pasta, and K. Wilhoit, "A security evaluation of ais automated identification system," in *Proceedings of the 30th ACM Annual Computer Security Applications Conference (ACSAC)*, 2014.

[26] RTCA Inc., *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)*, Dec. 2011.

[27] Y.-C. Tseng, S.-L. Wu, W.-H. Liao, and C.-M. Chao, "Location awareness in ad hoc wireless mobile networks," *Computer*, vol. 34, no. 6, Jun. 2001.

[28] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, May 2003.

[29] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proceedings of the 2nd ACM Workshop on Wireless Security (WiSe)*, Sep. 2003.

[30] S. Brands and D. Chaum, "Distance-bounding protocols," in *Advances in Cryptology*, workshop on the theory and application of cryptographic techniques (eurocrypt '93) ed. Springer Berlin Heidelberg, 1994, vol. 765.

[31] D. Singelee and B. Preneel, "Location verification using secure distance bounding protocols," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference (MASS)*, Nov. 2005.

[32] S. Čapkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, vol. 3, Mar. 2005.

[33] P. Perazzo, K. Ariyapala, M. Conti, and G. Dini, "The verifier bee: A path planner for drone-based secure location verification," in *Proceedings of the 16th IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2015.

[34] S. Čapkun, M. Čagalj, and M. Srivastava, "Secure localization with hidden and mobile base stations," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2006.

[35] M. Strohmeier, V. Lenders, and I. Martinovic, "Lightweight location verification in air traffic surveillance networks," in *Proceedings of the 1st ACM Workshop on Cyber-Physical System Security (CPSS)*, Oct. 2015.

[36] R. Baker and I. Martinovic, "Secure location verification with a mobile receiver," in *Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy (CPS-SPC)*, Oct. 2016.

[37] S. Čapkun and J.-P. Hubaux, "Securing position and distance verification in wireless networks," École polytechnique fédérale de Lausanne (EPFL), Tech. Rep., 2004.

[38] J. T. Chiang, J. J. Haas, and Y.-C. Hu, "Secure and precise location verification using distance bounding and simultaneous multilateration," in *Proceedings of the 2nd ACM Conference on Wireless Network Security (WiSec)*, Mar. 2009.

[39] L. Hu and D. Evans, "Using Directional Antennas to Prevent Wormhole Attacks," in *Network and Distributed System Security Symposium (NDSS)*, Feb. 2004.

[40] A. Vora and M. Nesterenko, "Secure location verification using radio broadcast," *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 4, Oct. 2006.

[41] J. Luo, H. V. Shukla, and J.-P. Hubaux, "Non-interactive location surveying for sensor networks with mobility-differentiated toa," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, Apr. 2006.

[42] R. Calvo-Palomino, F. Ricciato, D. Giustiniano, and V. Lenders, "LTESS-track: A Precise and Fast Frequency Offset Estimation for Low-cost SDR Platforms," in *Proceedings of the 11th ACM Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WINTECH)*, Oct. 2017.

[43] D. Adamy, *EW 101: A First Course in Electronic Warfare*, ser. Artech House Radar Library. Artech House, 2001.

[44] H. Li, D. Hestenes, and A. Rockwood, *Generalized Homogeneous Coordinates for Computational Geometry.* Springer Verlag, 2001.

[45] D. Moser, P. Leu, V. Lenders, A. Ranganathan, F. Ricciato, and S. Čapkun, "Investigation of multi-device location spoofing attacks on air traffic control and possible countermeasures," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom).* ACM, Oct. 2016.

[46] T. Schmid, "Time in wireless embedded systems," Ph.D. dissertation, University of California, Los Angeles, 2009.

[47] T. H. Witte and A. M. Wilson, "Accuracy of WAAS-enabled GPS for the determination of position and speed over ground," *Journal of Biomechanics*, vol. 38, no. 8, 2005.

[48] R. H. D. Townsend, "Fast calculation of the lomb-scargle periodogram using graphics processing units," *The Astrophysical Journal Supplement Series*, 2010.

[49] A. Amar and A. J. Weiss, "Localization of narrowband radio emitters based on doppler frequency shifts," *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5500–5508, 2008.

[50] "Minimum aviation system performance standards for Automatic Dependent Surveillance – Broadcast (ADS-B)," RTCA, Inc., Tech. Rep. DO-242A (including Change 1), Dec. 2006.

[51] "Minimum operational performance standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance – Broadcast," RTCA, Inc, Tech. Rep. DO-282B, Dec. 2011.

[52] *International Standards and Recommended Practices, Annex 10: Aeronautical Telecommunications*, 4th ed., International Civil Aviation Organization (ICAO), 2007, Volume IV: Surveillance and Collision Avoidance Systems.

[53] "Minimum operational performance standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)," RTCA, Inc., Tech. Rep. DO-260B (with Corrigendum 1), Dec. 2011.

[54] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, mar 1983.

[55] M. Strohmeier, M. Schäfer, R. Pinheiro, V. Lenders, and I. Martinovic, "On Perception and Reality in Wireless Air Traffic Communication Security," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, 2016.

[56] M. Strohmeier, "Security in next generation air traffic communication networks," Ph.D. dissertation, University of Oxford, 2016.

[57] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, vol. 4, 2011.

[58] N. Marz and J. Warren, *Big Data: Principles and Best Practices of Scalable Real-time Data Systems.* Manning Publications Co., Apr. 2015.

[59] M. Strohmeier, M. Schäfer, M. Fuchs, V. Lenders, and I. Martinovic, "OpenSky: A Swiss Army Knife for Air Traffic Security Research," in *Proceedings of the 34th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2015.

[60] M. Schäfer, M. Strohmeier, M. Smith, M. Fuchs, V. Lenders, M. Liechti, and I. Martinovic, "OpenSky Report 2017: Mode S and ADS-B Usage of Military and other State Aircraft," in *IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, Sep. 2017.

[61] R. M. Trim, "Mode S: an introduction and overview," *Electronics Communication Engineering Journal*, vol. 2, no. 2, April 1990.

[62] P. Park and C. Tomlin, "Investigating Communication Infrastructure of Next Generation Air Traffic Management," in *IEEE/ACM International Conference on Cyber-Physical Systems (ICCPS)*, Apr. 2012.

[63] W. H. Harman and M. J. Brennan, "Beacon Radar and TCAS Interrogation Rates: Airborne Measurements in the 1030 MHz Band," MIT Lincoln Laboratory, Tech. Rep., May 1996, prepared for the Federal Aviation Administration.

[64] D. Esler, "Global Advance Of ADS-B," accessed December 11, 2017. [Online]. Available: http://aviationweek.com/connected-aerospace/global-advance-ads-b

[65] Federal Aviation Administration, "Automatic Dependent Surveillance—Broadcast (ADS-B) Out Performance Requirements To Support Air Traffic Control (ATC) Service; Final Rule," *Federal Register*, vol. 75, no. 103, May 2010, 14 CFR Part 91.

[66] NEASCOG CNS Security Workshop, "NEASCOG Security Statement," 2013.

[67] Asia and Pacific Office of the International Civil Aviation Organization (ICAO), *ADS-B Implementation and Operations Guidance Document*, 7th ed., Sep. 2014.

Curriculum Vitae

Matthias Schäfer

Education

2013 - 2018	**PhD** in *Computer Science* Technische Universität Kaiserslautern, Germany Thesis: "*Mobility Improves the Security of Location Awareness in Wireless Networks*"
2010 - 2013	**Master of Science** in *Computer Science* Technische Universität Kaiserslautern, Germany Minor: business studies and economics Thesis: "*Design and Analysis of VeriFly - A Trajectory Verification Method based on RSS sampling*"
2006 - 2010	**Bachelor of Science** in *Computer Science* Technische Universität Kaiserslautern, Germany Minor: business studies and economics Thesis: "*Implementierung einer funkbasierten Schlüsselgenerierung für drahtlose Sensornetzwerke*"

Professional Experience

since 10/2018	**Lecturer** at *Department of Computer Science* Technische Universität Kaiserslautern, Germany Lecture: Network Security
since 1/2015	**Co-founder & Board Member** of the *OpenSky Network* Burgdorf, Switzerland Non-profit association supporting aviation research `https://opensky-network.org`

since 5/2014	**Founder & Directing Manager** of *SeRo Systems GmbH* Kaiserslautern, Germany Big Data and Security for Air Traffic Surveillance `https://sero-systems.de`
4/2013 - 9/2018	**Teaching Assistant** at *Distributed Computer Systems Lab* Technische Universität Kaiserslautern, Germany Lectures: Communication Systems, Network Security, Protocols and Algorithms for Network Security
1/2013 - 3/2013	**Internship** at *armasuisse W+T* Thun, Switzerland Information technology and cyberspace group
4/2012 - 11/2012	**Visiting researcher** at *University of Oxford* Oxford, United Kingdom Department of Computer Science Host: Prof. Dr.-Ing. Ivan Martinovic
11/2011 - 3/2012	**Internship** at *armasuisse W+T* Thun, Switzerland Information technology and cyberspace group
2009 - 2012	**Student Assistant** at *Embedded Systems Group* Technische Universität Kaiserslautern, Germany Research in the area of synchronous programming
2008 - 2011	**Tutor** at *Department of Computer Science* Technische Universität Kaiserslautern, Germany Lectures: Computer Systems 1 and 2

Honors & Awards

2017	**Cyber Award** 1st place for outstanding scientific contributions

	Issued by armasuisse, Thun, Switzerland Paper: *Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks*
2016	**Cyber Award** 3rd place for outstanding scientific contributions Issued by armasuisse, Thun, Switzerland Paper: *Secure Motion Verification using the Doppler Effect*
2015	**Best Paper Award** at the IEEE/AIAA Digital Avionics Systems Conference Paper: *OpenSky: A Swiss Army Knife for Air Traffic Security Research*
2015	**Cyber Award** 1st place for outstanding scientific contributions Issued by armasuisse, Thun, Switzerland Paper: *Secure Track Verification*
2014	**Best Demo Award** 1st runner up at the *ACM/IEEE Conference on Information Processing in Sensor Networks* Demo: *Opensky: A Large-scale ADS-B Sensor Network for Research*
2014	**Student Travel Grant** for the *ACM Conference on Security and Privacy in Wireless and Mobile Networks*
2011	**Security Award** of the Swiss Department of Defense Issued by IOS, Berne, Switzerland
2011	**Best Tutor Award** for the best result in the student course evaluation Issued by Fachschaft Informatik, Kaiserslautern, Germany Lecture: *Computer Systems 2*
2009	**Best Tutor Award** for the best result in the student course evaluation

Issued by Fachschaft Informatik, Kaiserslautern, Germany
Lecture: *Computer Systems 1*

Publications

Peer-reviewed Conference Contributions

1. **Matthias Schäfer**, Vincent Lenders, Ivan Martinovic: "Experimental Analysis of Attacks on Next Generation Air Traffic Communication", in *Proceedings of the 11th International Conference on Applied Cryptography and Network Security (ACNS)*, Springer, 2013

2. **Matthias Schäfer**, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, Matthias Wilhelm: "Bringing Up OpenSky: A Large-scale ADS-B Sensor Network for Research", in *Proceedings of the 13th IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2014

3. Martin Strohmeier, **Matthias Schäfer**, Markus Fuchs, Vincent Lenders, Ivan Martinovic: "OpenSky: A Swiss Army Knife for Air Traffic Security Research", in *Proceedings of the 34th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2015

4. **Matthias Schäfer**, Vincent Lenders, Jens Schmitt: "Secure Track Verification", in *Proceedings of the 36th IEEE Symposium on Security and Privacy (S&P)*, 2015

5. **Matthias Schäfer**, Patrick Leu, Vincent Lenders, Jens Schmitt: "Secure Motion Verification using the Doppler Effect", in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2016

6. Martin Strohmeier, Matthew Smith, **Matthias Schäfer**, Vincent Lenders, Ivan Martinovic: "Assessing the Impact of Aviation Security on Cyber Power", in *Proceedings of the 8th NATO CCD COE International Conference on Cyber Conflict (CyCon)*, 2016

7. **Matthias Schäfer**, Martin Strohmeier, Matthew Smith, Markus Fuchs, Rui Pinheiro, Vincent Lenders, Ivan Martinovic: "OpenSky Report 2016: Facts and Figures on SSR Mode S and ADS-B Usage", in *Proceedings of the 35th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2016

8. Martin Strohmeier, Matt Smith, **Matthias Schäfer**, Vincent Lenders, Ivan Martinovic: "Crowdsourcing Security for Wireless Air Traffic Communications", in *Proceedings of the 9th NATO CCD COE International Conference on Cyber Conflict (CyCon)*, 2017

9. **Matthias Schäfer**, Martin Strohmeier, Matthew Smith, Markus Fuchs, Vincent Lenders, Marc Liechti, Ivan Martinovic: "OpenSky Report 2017: Mode S and ADS-B Usage of Military and other State Aircraft", in *Proceedings of the 36th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2017

10. Kai Jansen, **Matthias Schäfer**, Daniel Moser, Vincent Lenders, Christina Pöpper, Jens Schmitt: "Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks", in *Proceedings of the 39th IEEE Symposium on Security and Privacy (S&P)*, 2018

11. Roman Trüb, Daniel Moser, **Matthias Schäfer**, Rui Pinheiro, Vincent Lenders: "Monitoring Meteorological Parameters With Crowdsourced Air Traffic Control Data", in *Proceedings of the 17th IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2018

12. Martin Strohmeier, Matthew Smith, Daniel Moser, **Matthias Schäfer**, Vincent Lenders, Ivan Martinovic: "Utilizing Air Traffic Communications for OSINT on State and Government Aircraft", *Proceedings of the 10th NATO CCD COE International Conference on Cyber Conflict (CyCon)*, 2018

13. Martin Strohmeier, Anna Niedbala, **Matthias Schäfer**, Vincent Lenders and Ivan Martinovic: "Surveying Aviation Professionals on the Security of the Air Traffic Control System", in *International Workshop on Cyber Security for Intelligent Transportation Systems (CSITS)*, 2018

14. **Matthias Schäfer**, Martin Strohmeier, Matthew Smith, Markus Fuchs, Vincent Lenders and Ivan Martinovic: "OpenSky Report 2018: Assessing the Integrity of Crowdsourced Mode S and ADS-B Data", in *Proceedings of the 37th IEEE/AIAA Digital Avionics Systems Conference (DASC)*, 2018

Peer-reviewed Journal Articles

1. Martin Strohmeier, **Matthias Schäfer**, Vincent Lenders, Ivan Martinovic: “Realities and Challenges of NextGen Air Traffic Management: The Case of ADS-B”, in *IEEE Communications Magazine*, Volume 52, Issue 5, 2014
2. Martin Strohmeier, **Matthias Schäfer**, Rui Pinheiro, Vincent Lenders, Ivan Martinovic: “On Perception and Reality in Wireless Air Traffic Communication Security”, in *IEEE Transactions on Intelligent Transportation Systems*, Volume 18, Issue 6, 2016

Peer-reviewed Posters & Demos

1. **Matthias Schäfer**, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, Matthias Wilhelm: “OpenSky – A Large-scale ADS-B Sensor Network for Research” (Demo), in *IEEE/ACM International Symposium on Information Processing in Sensor Networks (IPSN)*, 2014
2. **Matthias Schäfer**, Vincent Lenders, Jens Schmitt: “Secure Path Verification using Mobility-Differentiated ToA” (Poster), in *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2014
3. **Matthias Schäfer**, Daniel Berger, Vincent Lenders, Jens Schmitt: “Security By Mobility in Location and Track Verification” (Poster), in *ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, 2015
4. Kai Jansen, **Matthias Schäfer**, Vincent Lenders, Christina Pöpper, Jens Schmitt: “Localization of Spoofing Devices using a Large-scale Air Traffic Surveillance System” (Poster), in *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2017

Other

1. Martin Strohmeier, **Matthias Schäfer**, Ivan Martinovic: “Challenges in NextGen Air Traffic Management” (Invited Poster), in *Network and Distributed System Security Symposium (NDSS)*, 2014